Künstliche Intelligenz und Daten: Eine Evaluation softwarebasierter militärischer Informationsgewinnung

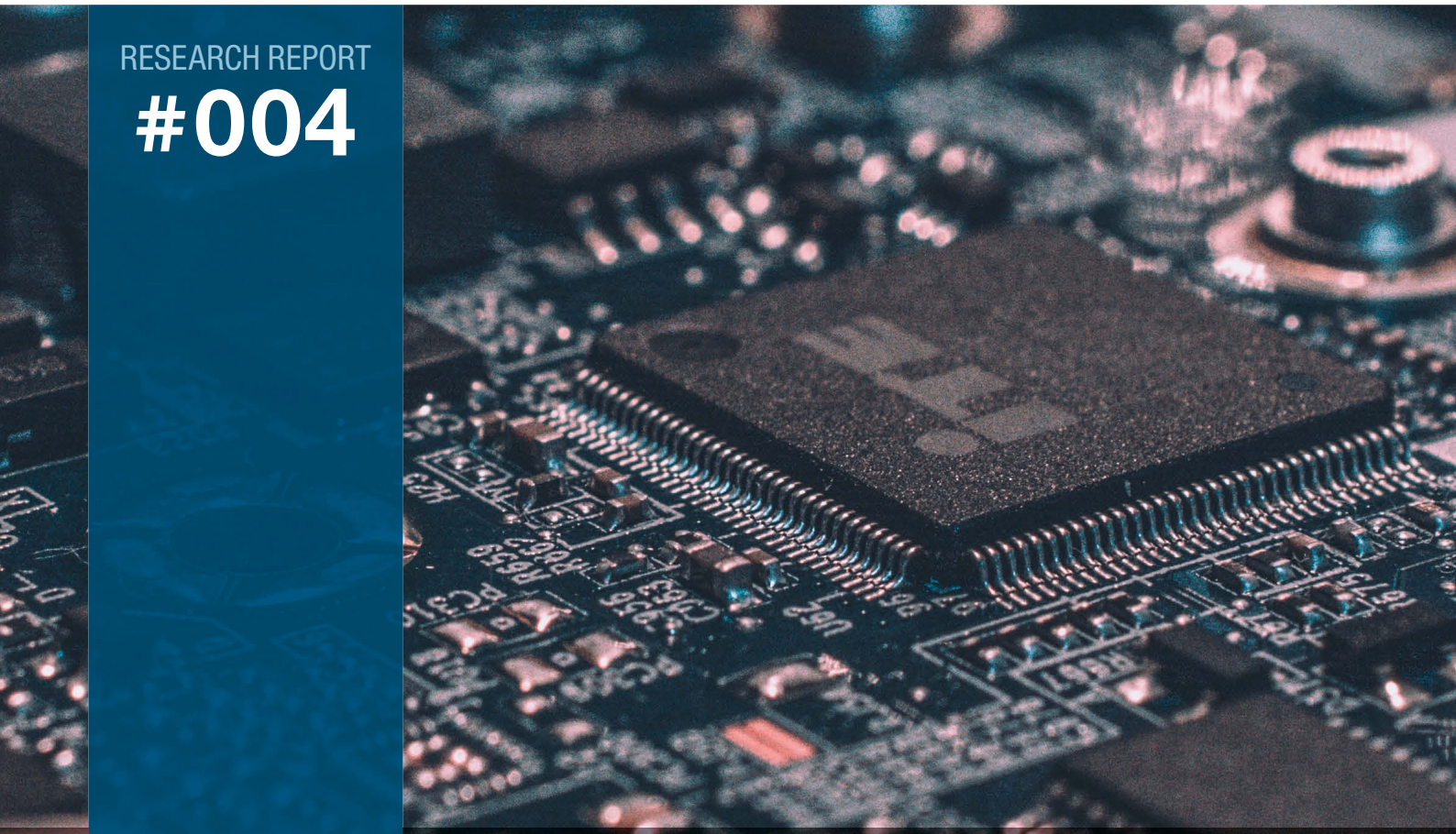
Erz, Hendrik

Veröffentlichungsversion / Published Version
Forschungsbericht / research report

Empfohlene Zitierung / Suggested Citation:
Erz, H.r. (2020). *Künstliche Intelligenz und Daten: Eine Evaluation softwarebasierter militärischer Informationsgewinnung.* (IFSH Research Report, 004). Hamburg: Institut für Friedensforschung und Sicherheitspolitik an der Universität Hamburg (IFSH). https://doi.org/10.25592/ifsh-research-report-004

IFSH
Institut für Friedensforschung
und Sicherheitspolitik
an der Universität Hamburg

RESEARCH REPORT
#004



Künstliche Intelligenz und Daten:

Eine Evaluation softwarebasierter militärischer Informationsgewinnung

HENDRIK ERZ | 06/2020

Inhalt

Zusammenfassung

Künstliche Intelligenz ist mittlerweile fester Bestandteil rüstungskontrollpolitischer Diskussionen im internationalen Raum. Doch das Aufkommen von künstlicher Intelligenz (KI) als Teil militärischer strategischer und taktischer Überlegungen stellt einerseits lang geltende Gewissheiten in Frage und offenbart andererseits eine tiefe Unsicherheit, was mit KI in Zukunft möglich sein wird. Um zu einer stabilen Grundlage dieses Diskurses beizutragen, analysiert dieser IFSH Research Report den derzeitigen softwaretechnologischen Entwicklungsstand – und zwar nah an den technologischen Realitäten, welche die militärischen Einsatzmöglichkeiten bestimmen. Zentrale Triebkraft von Software sind dabei Daten, die im Einsatzgebiet generiert werden und anschließend eine Vielzahl an Stationen durchlaufen, bis sie letztlich in die militärische Entscheidungsfindung einfließen. Anhand dieses Weges analysiert dieser Report die involvierten Algorithmen, Sensoren und weitere Technologien, die zur Informationsgewinnung in der digitalen Kriegsführung beitragen. Der Report schließt mit einer ersten Einschätzung, mit welchen militärischen Trends die internationale Gemeinschaft in den kommenden Jahren konfrontiert sein könnte und wo sich ein genauerer Blick, auch mit Bezug auf Rüstungskontrolle, lohnt.

Keywords: Artificial Intelligence, Machine Learning, Data Analysis, Network-Centric Warfare, Cybersecurity

Förderung

Dieser Research Report entstand im Rahmen des von der Deutschen Stiftung Friedensforschung (DSF) geförderten Forschungsvorhabens *„Algorithmen und Künstliche Intelligenz als Game Changer? Moderne Waffensysteme zwischen Erwartung und Wirklichkeit“* (Projektnummer FNT02/02–2018). Das Projekt untersucht die sicherheitspolitischen Implikationen von Softwaretechnologien. Handelt es sich bei ihnen um einen „Game Changer“, also einen kohärenten Technologieschub, der Elemente der Kriegführung revolutionieren wird? In dem Forschungsvorhaben soll auf Basis einer Technologie- und Inhaltsanalyse das interdisziplinäre Verständnis von Softwaretechnologien und ihres Einflusses auf die moderne Kriegführung sowie die internationale Sicherheit vertieft werden, um die Entwicklung adäquater Rüstungskontrollmechanismen zu unterstützen. Das Projekt ist am IFSH angesiedelt und wird von Christian Alwardt (Leitung), Hendrik Erz, Sylvia Kühne und Mirjam Limbrunner (SHK) bearbeitet.

1 Einleitung

Seit einiger Zeit sind Diskussionen um den Einfluss, die Möglichkeiten und Grenzen von künstlicher Intelligenz und Software fester Bestandteil rüstungskontrollpolitischer Diskussionen im internationalen Raum. Zahlreiche Akteure nehmen an dem Diskurs teil und geben Einschätzungen, Analysen und Prognosen zur Nutzbarkeit von Software für militärische Anwendungsszenarien ab. Maßgeblich geprägt wird der öffentliche Diskurs allerdings von politikwissenschaftlichen, sozialwissenschaftlichen oder rechtlichen Perspektiven. Dabei fallen zwei Dinge auf.

Zum einen ist die technologische Realität zahlreicher Anwendungsszenarien untererforscht. Während sich viele Autor*innen bereits mit „neuen Kriegen“ (Münkler 2004) und autonomen Waffensystemen („lethal autonomous weapon systems“, kurz LAWS) auseinandergesetzt haben (Chamayou 2015; Boulanin und Verbruggen 2017; Alwardt und Polle 2018), mangelt es immer noch an einem Verständnis dessen, was mit den heutigen Mitteln der Technik tatsächlich realisierbar ist, und welche zukünftigen Anwendungsszenarien aus den technologischen Möglichkeiten erwachsen.

Zum anderen verlangt Software völlig andere Kategorien als traditionelle Themen der Rüstungskontrolle. Anders als nukleare Waffen basiert sie nicht auf seltenen Rohstoffen und anders als konventionelle Waffen ist sie beliebig oft kopierbar. Durch ihre fehlende Materialität ist Software nur bedingt an physikalische Grenzen gebunden und nicht lokalisierbar. Daher können dieselben Algorithmen, die an einem Ort Sensordaten miteinander verknüpfen, an einem anderen Ort Material für virtuelle Karten zusammenstellen. Kurzum, es fehlen die Begriffe, um künstliche Intelligenz in rüstungskontrollpolitischen Kontexten besprechen zu können (Davis 2019: 2).

Ziel des DSF-geförderten Projektes „Algorithmen und künstliche Intelligenz als Game Changer?“ ist, das Verständnis um die Rolle und den Einfluss von Softwaretechnologien in der zukünftigen Kriegsführung zu vertiefen, die damit einhergehenden sicherheitspolitischen Implikationen auszuleuchten und neue Wege in der Rüstungskontrolle von Softwaretechnologien zu bereiten. Dieser Research Report konzentriert sich dabei auf den naturwissenschaftlich-technischen Aspekt und macht den Versuch, die Software zu untersuchen, die von Staaten in militärischen Operationen weltweit genutzt werden kann. Dadurch soll das Wissen erweitert werden, mit welchem rüstungskontrollpolitische Analysen von Software möglich werden.

Der vorliegende Report fokussiert sich auf drei spezifische Problemstellungen, die für den rüstungskontrollpolitischen Diskurs von besonderer Bedeutung sind. Erstens geht es diesem Report darum, den derzeitigen Stand der softwaretechnologischen Entwicklung zu skizzieren und erste kurz- und mittelfristige Trends abzuschätzen. Zweitens stehen die Softwareanwendungen selbst im Fokus: Wie effizient sind sie, was können sie tatsächlich leisten und welche Anforderungen stellen sie an die Hardware, auf der sie laufen? Drittens problematisiert dieser Report das Anwendungsspektrum von Software und schätzt ein, inwiefern sich aus der dargestellten Software militärische Anwendungsszenarien ergeben könnten. Mit dem so gewonnenen Wissen können sicherheitspolitische Risiken analysiert und nötige Schritte der Rüstungskontrolle abgewogen werden.

Um sich diesen drei Fragen anzunähern, ist es zunächst notwendig, die relevanten Kategorien von Softwareanwendungen zu bestimmen, das heißt, ihre operationalisierbaren Attribute herauszufiltern. Dazu gehören die verschiedenen Klassen von Algorithmen, ihre Hardware-Anforderungen sowie die in diese Prozesse involvierten Daten. Leider können die Hardwareanforderungen nicht exakt bestimmt werden, da einerseits die Forscher*innen für ihre Algorithmen nur wenige konkrete Angaben machen und andererseits die verfügbare Rechenleistung an Bord von mobilen Waffenplattformen wie Drohnen nicht bekannt ist. Fast sicher ist nur, dass mobile Plattformen über eine geringere Rechenleistung als festinstallierte Server in stationären Command & Control[1]-Stützpunkten verfügen, da erstere mit Einschränkungen in Größe, Gewicht und Energieverbrauch konfrontiert sind (vgl. ADLink 2017). Zur in die jeweiligen Prozesse involvierten Datenmenge lässt sich mehr sagen, doch die dem Militär zur Verfügung stehende Bandbreite zur Übertragung dieser kann trotz konkreter Zahlen nur geschätzt werden. Für diese Bereiche versucht dieser Report, auf Grundlage der öffentlich verfügbaren Informationen Einschätzungen abzugeben.

Dreh- und Angelpunkt von Software sind Daten, da das fundamentale Grundprinzip von jeglicher Software Algorithmen sind, die eine Eingabe (Input) erhalten und mittels dieser eine Ausgabe (Output) generieren. Zum einen sind das simple Konfigurationsparameter, mit welchen sich die Software einstellen lässt, ähnlich wie sich mittels eines Rads eine Herdplatte steuern lässt. Auf der anderen Seite – und dies ist die für diesen Report maßgebliche Form – stehen Daten, die mittels Sensoren über die Umwelt gesammelt werden und sich daher in Art und Menge erheblich von solchen Konfigurationsparametern unterscheiden. Diese Daten sind Fokus des vorliegenden Reports und anhand des Weges, wel-

chen sie durch die verschiedenen Stationen militärischer *situational awareness* nehmen, analysiert der Report die beteiligten Algorithmen und Systeme.

Ein Verständnis der militärischen Systemen zugrundeliegenden Softwaretechnologien kann dabei helfen, effektive Rüstungskontrolle solcher „weichen" Technologien auf den Weg zu bringen. Drei Besonderheiten von Software machen eine effektive Regulierung allerdings zu einer schwierigen Aufgabe. Erstens kann das Abändern weniger Zeilen Code aus einem Produkt rechtlich ein anderes machen. Zweitens weisen die diesen Technologien zugrundeliegenden Algorithmen inhärent Dual-Use-Charakteristiken auf[2] und sind durch wissenschaftlichen Austausch weltweit bekannt. Zudem kommt ein Großteil der militärisch eingesetzten KI aus Unternehmen, sodass die Staaten hier sehr vom privaten Sektor abhängig sind (vgl. Davis 2019: 13). Drittens ist durch die fehlende Materialität solcher Softwareprodukte das Nachverfolgen und damit die Verifikation von Exportbeschränkungen schwierig. Zwei Beispiele zeigen allerdings, dass mit einem ausreichenden Verständnis der Technologien Rüstungskontrolle sowie Verifikation durchaus möglich sind. So konnten Thorsten Schröder und Ulf Buermeyer (2019) mittels Computerforensik nachweisen, dass die deutsche Spionagesoftware FinSpy *nach* dem Inkrafttreten von Exportbeschränkungen in die Türkei gelangte, wo sie gegen Oppositionelle eingesetzt wurde. Außerdem nutzt ein Projekt des Lawrence Livermore National Laboratory Machine Learning, um in Datenströmen nach Aktivitäten von Waffenproliferation zu suchen (Davis 2019: 6). Die Ergebnisse dieses Research Reports könnten solchen Initiativen also helfen, Software in Debatten zur Rüstungskontrolle zu inkorporieren.

2 Technische Begrifflichkeiten und ihre Verwendung

Im vorliegenden Research Report werden einige Begriffe verwendet, deren Gebrauch entweder kontextabhängig bzw. unscharf ist, oder bei welchen sich die Verwendung im öffentlichen und im technischen Diskurs unterscheidet. Im Folgenden wird auf diese problematischen Fälle eingegangen und die Verwendung der Begriffe im Kontext dieses Reports klarifiziert.

Der Begriff „künstliche Intelligenz“ (KI) ist ein Oberbegriff für die in diesem Report untersuchten Technologien. Fast alle der untersuchten Algorithmen und Paradigmen lassen sich je nach Perspektive als KI bezeichnen. Doch während der Begriff KI vor allem verwendet wird, um im *Allgemeinen* über Fähigkeiten von Software zu sprechen, untersucht dieser Report *spezielle* Anwendungsfälle von Software. Ein weit verbreitetes Einführungswerk in „Artificial Intelligence“ von George Luger (2008) beginnt mit der Feststellung, dass, obwohl wir intelligentes Verhalten erkennen, wenn wir es sehen, es keine konkreten Anhaltspunkte dafür gibt, wie sich dieses definieren ließe (ebd.: 2–4; ähnlich auch Davis 2019: 2). Die IEEE-Studie zu „Ethically Aligned Design“ (IEEE Global Initiative on Ethics of Autonomous and Intelligent Systems 2019) erklärt, dass der Begriff in der Informatik hauptsächlich metaphorische Verwendung finde, während „general and popular discourse may not share in the same nuanced understanding“ (ebd.: 37–38). Das wird auch in den im Folgenden zitierten Artikeln und Büchern deutlich: Keines der Paper aus dem Bereich der Informatik verwendet den Begriff; dafür aber die übergreifenden Bücher und Veröffentlichungen des US-Verteidigungsministeriums und anderer militärischer Institutionen, die sich mehr mit Strategie und Taktik, weniger aber mit der konkreten Implementation von Software befassen.

Für ein Verständnis von Softwareanwendungen und insbesondere der Einschätzung der zur Verfügung stehenden Bandbreite ist auch wichtig, die Größenordnungen der entstehenden Daten im Blick zu behalten. Während die Begriffe Megabyte und Kilobyte den meisten Leser*innen bekannt sein dürften, wird vor allem im Kontext von Daten*übertragung*, das heißt Kommunikation, nicht von Byte, sondern von Bits gesprochen. Zum schnelleren Verständnis wurden alle Zahlen im vorliegenden Text in Byte umgerechnet.[3]

In wissenschaftlichen Publikationen, vornehmlich mit Bezug auf das Thema Sensorik, wird vielfach von *data fusion* gesprochen. Damit ist im engeren Sinne allerdings speziell die sogenannte *sensor fusion* gemeint, das heißt das Kombinieren von Sensordaten. Die Begriffe, die zur Beschreibung von Datenfusionierung benutzt werden, sind sehr schwammig und verfügen über keine einheitliche Definition (vgl. Elmenreich 2002: 2). In diesem Research Report werden zwei Formen von Fusion behandelt, nämlich *sensor fusion* und *information integration*. Diese werden in Anlehnung an Elmenreich wie folgt voneinander abgegrenzt: *Sensor fusion* bezeichnet das Zusammenfassen von Rohdaten mehrerer Sensoren direkt am *Anfang* der Informationskette, während *information integration* das Kombinieren von bereits vorverarbeiteten Daten bezeichnet, was vor

allem am *Ende* der Informationsgewinnung stattfindet (vgl. die Unterscheidung von *sensor fusion* und *information fusion* ebd.: 3). Eine weitere Problematik betrifft die mit der militärischen Software direkt verbundene Frage nach dem Grad der Autonomie von Waffensystemen, die seit 2013 auch auf der *Convention on Prohibitions or Restrictions on the Use of Certain Conventional Weapons (CCW)* diskutiert wird. Hier gibt es weder Klarheit darüber, was „echte" Autonomie umfasst, noch wie autonom aktuelle Waffensysteme überhaupt sind. Da eine „wirkliche" Autonomie derzeit noch nicht erreicht ist, wird im weiteren Verlauf auf den Begriff „Autonomie" verzichtet (vgl. Scharre und Horowitz 2015; Boulanin und Verbruggen 2017; IEEE Global Initiative on Ethics of Autonomous and Intelligent Systems 2019).

Zuletzt muss die Verwendung des Begriffes „neuronales Netzwerk" problematisiert werden. Seit der Erfindung des „Perzeptrons" durch Frank Rosenblatt in den 1950er Jahren hat sich der Begriff „neuronales Netzwerk" für die Weiterentwicklungen der Bilderkennung etabliert. Allerdings muss nach jahrzehntelanger Diskussion auch in der Wissenschaft konstatiert werden, dass dieser Begriff trotz seiner Verbreitung nicht korrekt ist (vgl. bereits Crick 1989 für einen ausführlichen Kommentar zur Bezeichnung „neuronales Netzwerk"). Was als „neuronales Netzwerk" bekannt ist, funktioniert nur bedingt wie ein tatsächliches (rhizomatisches, d.h. nicht-hierarchisches) Netzwerk, sondern eher wie eine „Pipeline", und auch die einzelnen Knotenpunkte innerhalb solcher „Netzwerke" haben nur wenig Gemeinsamkeiten mit Gehirnzellen (ebd.). Daher wird im Folgenden auf den Begriff „neuronales Netzwerk" verzichtet. Stattdessen wird der Begriff „modellbasierter Klassifizierer" verwendet, da dies der Funktionsweise dessen am nächsten kommt: Mittels eines Modells, welches durch das vorhergehende „Training" (also „machine learning", vgl. Kapitel 7.2) des Algorithmus' erzeugt wird, ist es dem Algorithmus möglich, Daten, die jenen aus der Trainingsphase ähnlich sind, korrekt in vorgegebene Kategorien zu klassifizieren.

Auf einer abstrakteren Ebene ist noch eine Frage offen, nämlich jener nach der Lokalisierung von Software. Bislang wurde konsistent von einem „Pfad" bzw. „Weg" gesprochen, welchen Daten von ihrer Erhebung bis zur Analyse in Command & Control-Stützpunkten nehmen. Doch diese Bezeichnung kann nur als Analogie verstanden werden. Denn während in vielen Erkundungs-Szenarien oder Kampfeinsätzen derzeit Daten tatsächlich auch über geographische Entfernungen übermittelt werden, können sämtliche Schritte auch auf einem einzigen Computer ausgeführt werden. Natürlich ist es aufgrund zahlreicher Gründe, die im Folgenden erläutert werden, für militärische Zwecke sinnig, die Informations-

gewinnung auf mobile Geräte wie Drohnen zu verteilen und die Analyse dieser Informationen in Rechenzentren zu zentralisieren, doch so wird es verständlich, weshalb dieser Report konsistent von „Einsatzgebiet“ sprechen wird und auf konkrete Ortsangaben verzichtet, da diese Terminologie prinzipiell einen weiteren Raum einschließt. Dies erklärt im Übrigen auch die Entscheidung, diesen „Pfad“ in Abbildung 1 (vgl. unten) in Schichten darzustellen.

3 Analyse softwaregestützter Informationsgewinnung – ein Konzept

Da es sich bei dem vorliegenden Research Report um eine explorative Analyse der potenziellen Nutzung von Software in militärischen Anwendungskontexten handelt, stellt sich hier insbesondere das Problem der Struktur. Während die *Domänen*, in welchen Software eingesetzt werden kann, bereits bestehen – beispielsweise *situational awareness, command & control, early warning* oder *safety* – sind die Kategorien, in welche sich Software sortieren lässt, bislang nur unzureichend ausgeleuchtet.

Um diesem Problem zu begegnen, nutzt dieser Report als Grundlage für die Bildung der Kategorien gemäß der eingangs formulierten Feststellung, dass der Dreh- und Angelpunkt von Software Daten sind, den Entstehungsweg militärisch nutzbarer Informationen vom Einsatzgebiet bis zur Analyse in Command & Control-Stützpunkten. Dieser Weg der Informationsgewinnung und die daraus resultierenden Kategorien werden nach einer Einführung in die Materialgrundlage im Folgenden vorgestellt.

3.1 MATERIALGRUNDLAGE

Um die zentralen Fragen mit Blick auf die genannten Ziele beantworten zu können, nutzt dieser Research Report drei Arten von Material. Zum einen bezieht er sich auf wissenschaftliche Publikationen, in welchen verschiedene Algorithmen

im Detail vorgestellt werden. Vorrangig werden hierfür Veröffentlichungen aus verschiedenen Zeitschriften des *Institute of Electrical and Electronics Engineers (IEEE)* und Preprints von arXiv herangezogen. Zweitens nutzt er Whitepaper aus der Industrie sowie Strategiepapiere des US-Militärs, um einschätzen zu können, wo und inwiefern diese Algorithmen militärisch genutzt werden könnten. Diese beiden Quellen werden drittens mit Medienberichten aus einschlägigen Portalen für militärische Entwicklungen wie beispielsweise *War on the Rocks* sowie Zeitungsartikeln ergänzt. Insbesondere mit Bezug auf mögliche militärische Anwendungen ist der vorliegende Research Report des weiteren stark mit Blick auf die USA verfasst, da einerseits die US-Streitkräfte außerordentlich offen im Umgang mit ihren Strategien sind und diese in zahlreichen Publikationen darstellen, und andererseits hier – anders als bei russischen oder chinesischen Initiativen – keine Sprachbarriere vorherrscht.

Im Zuge der Recherche haben sich drei Punkte herauskristallisiert, welche dafür sprechen, dass die hier vorgestellten Ergebnisse wahrscheinlich auf militärische Anwendungsszenarien übertragbar sind. Zum einen setzen militärische Akteure immer stärker auf sogenannte „off-the-shelf“[4] -Software, also kommerzielle Anwendungen, die nicht in erster Linie für das Militär entwickelt wurden. Zum anderen entwickelt vor allem das US-Militär selbst Software, die es unter Open Source[5]-Lizenzen zur Verfügung stellt. Drittens agiert das Militär nicht in einer von der restlichen Gesellschaft völlig abgetrennten Sphäre, weswegen kommerzielle Paradigmen – zum Beispiel „Sharding“[6] und verteilte Dateisysteme – auch vom Militär genutzt werden; genauso wie Smartphones zu weiten Teilen durch militärische Technologien ermöglicht wurden (vgl. Mazzucato 2014: 115–145).

So ist die Nutzung von Open Source zum Beispiel in der Initiative *Code.mil* ersichtlich, welche Angehörige der US-amerikanischen Streitkräfte ermuntern soll, ihren Software-Code unter einer Open Source-Lizenz zur Verfügung zu stellen. Das für die Programmierung von Zielerfassungsalgorithmen benutzte Framework[7] „The Tracker Component Library“ (Crouse 2017) ist nur ein Beispiel für die Open Source-Anstrengungen des US-Militärs.[8] Dieser Report kann also auf eine große Menge Open Source-Software und ziviler Forschung und Entwicklung (F&E) zurückgreifen, die auch das Militär nutzt; auf Entwicklungen im Bereich der Algorithmik, spezifisch Zielerkennungsalgorithmen oder der Fusionierung verschiedener Datenquellen.

3.2 MILITÄRISCHE INFORMATIONSGEWINNUNG: DATEN ALS BINDEGLIED

Um die Softwarelandschaft mit Fokus auf mögliche militärische Anwendungsszenarien zu scannen, konzentriert sich dieser Report auf das Nachverfolgen des Weges von Daten innerhalb militärischer Anwendungen von der Datensammlung im Einsatzgebiet bis zur Analyse und Speicherung. Da die Datenverarbeitung zentrales Bindeglied für Software ist, hat sich dies als plausible Strukturierung erwiesen. Aus dieser ersten Eingrenzung ergeben sich fünf Oberkategorien, in welche sich die mögliche militärische Nutzung von Software einordnen lässt. Diese sind Sensorik, eingebettete und integrierte Systeme, Kommunikation, Datenprozessierung und -analyse sowie Datenspeicherung und -verwaltung.

Diese fünf Oberkategorien sind integraler Bestandteil der militärischen *situational awareness*. Ausgehend von der Datengenerierung mittels Sensoren bis hin zur Analyse in Command & Control-Stützpunkten zeichnet dieser Research Report also maßgeblich den Prozess nach, mit welchem die militärische Informationsgewinnung funktioniert. Auch in Handbüchern der US-Streitkräfte wird mit Bezug auf „network-centric warfare“ und informationeller Kriegsführung der Fokus auf Informationsgewinnung gelegt (vgl. Alberts 1999: 89).

Gleichzeitig lässt sich dieser Prozess jedoch auch umgekehrt lesen: Ausgehend von analysierten Daten aus Command & Control-Stützpunkten verläuft das militärische *decision making* häufig entlang ähnlicher Pfade, weshalb diese Oberkategorien ebenfalls integraler Bestandteil der militärischen Steuerung sind. Dieser Prozess wird mit Bezug auf die „network-centric warfare“ immer wieder sichtbar (vgl. U.S. Army 2003; Alberts 1999). Auch geht die Informationsgewinnung häufig von denselben Punkten aus, an welchen die darauf folgende Befehlskette endet (viele Drohnen der US-Streitkräfte beispielsweise sind mittlerweile auch waffenfähig und sammeln nicht mehr nur Informationen).

Bei diesen Kategorien handelt es sich nicht um abschließende Definitionen, sondern vielmehr um zielführende Abgrenzungen, da sich beispielsweise eingebettete[9] und integrierte Systeme nicht trennscharf abgrenzen lassen (vgl. Kapitel 5). Die Sinnhaftigkeit dieser Kategorien lässt sich anhand gängiger schematischer Darstellungen von militärischen Netzwerken auf dem Schlachtfeld belegen, die vereinfacht in Abb. 1 wiedergegeben werden (vgl. Alberts 1999: 89; U.S. Army 2003). Die Unterteilung in Datenspeicherung und Datenverarbeitung ist aus den Erläuterungen des Militärs nicht explizit ersichtlich, ergibt sich aber aus Notwen-

Abbildung 1: Prozess militärischer Informationsgewinnung

Quelle: eigene Darstellung.

digkeiten in der Informationstechnologie. Dort gibt es grundsätzlich getrennte Systeme für einerseits die Analyse, und andererseits die Speicherung und Aufbewahrung von Daten.

Insbesondere für ein Verständnis der Informationsverarbeitung im Command & Control wird auf gängige Praxen der Softwareindustrie in der Datenverarbeitung zurückgegriffen. Grund dafür ist, dass Command & Control in zahlreichen Abbildungen als eine „black box" dargestellt wird und die datenverarbeitenden Prozesse nicht ersichtlich sind. Während die genauen Vorgänge der Datenanalyse der Geheimhaltung unterliegen, folgt dieser Report der naheliegenden Vermutung, dass das Militär hier auf bewährte Methoden setzen wird; auch und insbesondere durch die Kooperationen mit Software-Unternehmen (die Nutzung von „Off-the-shelf"-Software sowie der Einkauf von Cloud Computing-Ressourcen für die US-Streitkräfte bspw. über das JEDI-Projekt).

3.3 DATENPFADE: EINFÜHRUNG IN DIE ZENTRALEN KATEGORIEN

Bevor Software mit Daten arbeiten und diese analysieren kann, ist es zunächst erforderlich, Informationen aus der realen Welt in digitale, maschinenlesbare Daten umzuwandeln, was mittels Sensorik geschieht (vgl. Kapitel 4). Algorithmen helfen dabei, die Daten solcher Sensoren, beispielsweise Kameras oder optische Abstandsmesser, einerseits auszuwerten, andererseits aber auch Störsignale heraus zu filtern. Die so – beispielsweise mittels Drohnen oder mobiler Radarstationen – gewonnen Daten können dann weiterverarbeitet werden.

Dabei ist allerdings insbesondere mit Blick auf Doktrinen wie „network-centric warfare" oder, aktueller, „mosaic warfare" (vgl. Jensen und Paschkewitz 2019), zu beachten, dass mit dem Wunsch, die Waffensysteme modular zu halten, d.h. unabhängig voneinander, auch gewisse Beschränkungen in der Prozessionsleistung einhergehen. Sollen kleine, mobile Waffensysteme wie Drohnen bereits Datenanalyse ausführen, ist zu beachten, dass diese über weniger Rechenleistung verfügen, als die Serversysteme, welche in stationären Command & Control-Stützpunkten vorgehalten werden. Dies wird in Form der Unterscheidung in *embedded systems* („Eingebettete Systeme") versus *integrated systems* („Integrierte Systeme", im US-Militärjargon „System of Systems") Fokus des darauf folgenden Kapitels sein.

Unter der Kategorie Kommunikation wird daran anschließend untersucht, wie viele Daten mit derzeitigen Technologien vom Einsatzgebiet aus gesendet werden können, wobei das kommerzielle Internet ausgeklammert wird, da hierbei bereits sehr hohe Geschwindigkeiten erreicht werden können, die (noch) keinen nennenswerten Flaschenhals darstellen.[10] Viel wichtiger sind Kommunikationskanäle im Einsatzgebiet selbst sowie kommerzielle Satellitensignale, da diese einerseits durch (feindliche) Störsignale beeinträchtigt werden können und andererseits der Sicherheitsaspekt die Bandbreite dieser Kanäle zusätzlich reduzieren kann.

Datenprozessierung und -analyse ist die nächste Kategorie, unter welcher zentral Algorithmen analysiert werden, die für Zielerfassung, Raketensteuerung und Überwachung einsetzbar sind. Hierbei wird die SWaP-Restriktion („Size, Weight, and Power“, also Größe, Gewicht und Energieverbrauch) von eingebetteten Systemen eine Rolle spielen, um abzuschätzen, welche Daten bereits vorverarbeitet und dadurch komprimiert werden können. Es ist anzunehmen, dass das Militär möglichst wenig Daten über Kommunikationskanäle übertragen möchte, da die militärischen Datenlinks nur über eine geringe Bandbreite verfügen (vgl. Kapitel 6).

In einer letzten Kategorie, Datenspeicherung und -verwaltung, verlässt dieser Research Report dann das Gebiet der mobilen Waffensysteme und betritt das ausschließliche Terrain von stationären Rechenzentren in Command & Control-Stützpunkten, denn die gesammelten und verarbeiteten Daten müssen schließlich gespeichert werden, um Analysen zu ermöglichen und die Systeme mit den neuen Daten zu verbessern. Hierzu werden aktuell verwendete Paradigmen vorgestellt, mit denen das Verwalten großer Datenmengen („big data“) sowohl in Bezug auf Speicher- wie auch Zeitressourcen möglich wird. Dieses Kapitel fokussiert sich weniger auf konkrete Implementationen, sondern mehr auf Ansätze zur Lösung der Probleme, die mit „big data“ einhergehen.

4 Sensorik

Sensoren erlauben es, Daten über die Umgebung zu sammeln, damit Software diese auswerten kann. Da es im militärischen Kontext mit Bezug auf Informationssysteme (INFOSYS im Field Manual der U.S. Army, vgl. 2003) hauptsächlich

um *situational awareness* geht, also das frühzeitige Erkennen und Überwachen von Feindbewegungen, sind Sensoren integraler Bestandteil der Beschäftigung mit Software im militärischen Kontext. Im Folgenden interessieren besonders die *Auflösung* der Daten, die *Menge* der Daten sowie die *Präzision* der Sensoren, da beispielsweise Infrarotsensoren mit geringer Auflösung bereits wichtige Informationen gewinnen können, sich mit der Datenflut von hochauflösenden Überwachungssystemen *(Wide Area Motion Imagery, WAMI)* aber ganz andere Probleme für die militärische Aufklärung ergeben.

Bevor Software in der Lage ist, Analysen durchzuführen, welche das Militär für seine Entscheidungsfindung nutzen möchte, muss die Realität, welche sich im Einsatzgebiet vorfindet, als maschinenlesbare Informationen bereitgestellt werden. Dabei müssen Algorithmen fundamentale Probleme lösen, welche sich menschlichen Kombattanten nicht stellen: sie sind nicht in der Lage, Kontextinformationen ohne Hilfe in die Auswertung mit einfließen zu lassen und können nicht ohne Vorarbeit mehrere Datenquellen miteinander verbinden. Da aber Maschinen nicht an die menschlichen Sinnesorgane gebunden sind, können sie andererseits eine Vielzahl an Informationen erfassen, welche für Menschen unsichtbar sind.

4.1 OPTISCHE KAMERAS

Optische Kameras sind heutzutage allgemein verbreitet. Viele Menschen verfügen über eine im Smartphone eingebaute Kamera. Kameras waren immer schon intrinsischer Teil von „situational awareness“ und Überwachung (vgl. Shi et al. 2012: 1; für einen historischen Überblick Meyer 2019). Kameradaten werden zunehmend von automatischen Systemen analysiert. Der Hauptnutzen von Kameras im militärischen Bereich ist die Überwachung, also die Wahrung von *situational awareness*. Hierfür werden zunehmend auf Drohnen montierte Kamerasysteme verwendet.

Neben einzelnen Kameras gibt es sogenannte *Wide Area Motion Imagery*-Systeme *(WAMI)*; die US-Navy bezeichnet die gleichen Systeme als *Broad Area Maritime Imagery (BAMI)*. Dabei handelt es sich um auf Drohnen montierte Kamerasysteme, welche die Videos von mehreren nebeneinander platzierten Kameras zusammenführen, um ein wesentlich größeres Blickfeld („field of view“, FOV) zu erhalten. Bereits Grégoire Chamayou warnte vor den Überwachungs-Fähigkei-

ten von „Gorgon Stare“, dem bekanntesten WAMI-System, (2015: 43), welches pro Bild rund 1,8 Milliarden Pixel erzeugen kann (vgl. Leon 2019).

Optische Kameras werden vom Militär vornehmlich auf Drohnen eingesetzt, um aus der Vogelperspektive einen weiträumigen Blick über das Einsatzgebiet zu erhalten. Die so gewonnenen Daten werden mittels Internet in die Steuerzentralen der Drohnen gesendet (vgl. Kapitel 6), wo sie entweder von menschlichen Analyst*innen oder automatischer Software ausgewertet werden. Für die Zielerkennung werden sie nicht benutzt, da hierfür Infrarotkameras genutzt werden (vgl. weiter unten).

Zudem wird derzeit verstärkt im Bereich *Hyperspectral Imagers* (*HSI)*, vgl. weiter unten, geforscht, die neben dem sichtbaren Lichtspektrum zusätzlich unsichtbare Spektren erfassen, also mehr Informationen bereitstellen, als herkömmliche Kameras. HSIs werden aufgrund ihrer Vorteile (vgl. Knight 2019) womöglich in Zukunft mehr und mehr für die militärische Aufklärung verwendet; dass sie optische Kameras allerdings ersetzen werden, erscheint unwahrscheinlich. Mit fortschreitenden Entwicklungen im Bereich von Bilderkennungsalgorithmen werden Kameras nämlich zunehmend für die automatische Navigation von unbemannten Fahrzeugen genutzt (z.B. bei DroNet, vgl. Loquercio et al. 2018). Der US-amerikanische Automobilkonzern Tesla teilte 2019 mit, man könne mittlerweile mit Kamerabildern ähnlich präzise die Abstände zu entfernten Objekten messen, wie mit LiDAR-Sensoren (vgl. Field 2020; zu LiDAR-Sensoren siehe unten), was von einer aktuellen Studie bestätigt wird (vgl. Crowe 2019; Wang et al. 2020). Dieser Trend wird mutmaßlich dazu führen, dass Kameras in Zukunft mehr und mehr zur Navigation eingesetzt werden und hier zur ernstzunehmenden Konkurrenz für die teureren LiDAR werden (vgl. Crowe 2019).

Die Achillesferse optischer Kameras ist allerdings die Art und Menge der Daten. Zum einen produzieren hochauflösende Kameras eine enorme Datenmenge. Bei der Umwandlung der Sensordaten in ein Digitalformat mittels des H.264-Codec[11] (vgl. für einen Überblick Wiegand et al. 2003: 561–562) entstehen bis zu 30 Megabyte Daten pro Sekunde (ebd.: 573), sodass bestehende Kommunikationskanäle an ihre Grenzen stoßen. Militärische Datenlinks wie beispielsweise Link 16 können maximal rund 125 Kilobyte pro Sekunde übertragen (Martinez-Ruiz et al. 2010). Zum anderen sind Kamerabilder nicht maschinenlesbar und müssen daher analysiert werden, was nur mittels modellbasierter Klassifizierer Sinn macht (vgl. Kapitel 7).

4.2 INFRAROTKAMERAS

Im Gegensatz zu regulären Kameras nehmen Infrarotkameras nur das Infrarotspektrum, d.h. Wärmeabstrahlung, auf. Infrarotkameras haben ein kleines FOV von nur 640x512 Pixeln (Cao et al. 2015: 9; Wu et al. 2019: 16; Javadnejad et al. 2020: 3), also erheblich geringer als reguläre Kameras. Daher kommt es, dass im militärischen Kontext meist von FLIR oder SLIR gesprochen wird, das heißt „forward-looking" bzw. „side-looking" Infrarot. Infrarotsensoren müssen also mehr bewegt werden, um das gleiche Bildmaterial zu generieren. Deshalb werden sie – je nach Szenario – seitwärts (was auch unterhalb eines Flugzeuges mit einbezieht) oder nach vorne ausgerichtet.

Infrarotsensoren verfügen noch über eine Legacy-Auflösung[12], die sich seit langem nicht erhöht hat. Gemäß dem hier abgebildeten Wissensstand werden sie anders als die mit ihren Daten arbeitenden Algorithmen nicht aktiv beforscht. Dies scheint darin begründet zu sein, dass aktuelle Algorithmen auch mit dieser begrenzten Auflösung sehr hohe Präzision erzielen können (vgl. den Abschnitt 7.1 über Zielerfassungsalgorithmen). Ihre geringe Auflösung scheint für militärische Zwecke also nicht nachteilig zu sein, sondern eher von Vorteil, da die geringe Datenmenge militärische Kommunikationskanäle entlasten könnte.

Infrarotsensoren werden seit jeher auch auf Satelliten eingesetzt. Dort haben sie teils eine räumliche Auflösung von einem Kilometer je Pixel. Allerdings sind Landsat 7 und 8 präziser. Sie können bis zu 30 Meter je Pixel auflösen (vgl. Javadnejad et al. 2020: 1–2). Zusätzlich zu dieser geringen *Auflösung* verfügen Infrarotsensoren über eine geringe *Präzision*: „in contrast to visual images, the images obtained from an IR sensor have extremely low signal to noise ratio (SNR), which results in limited information for performing detection tasks" (Yilmaz, Shafique und Shah 2003: 623). Signal-to-Noise-ratio beschreibt dabei das Verhältnis von verwendbaren Daten (Signale) zu Störgeräuschen (Noise).[13]

4.3 HYPERSPECTRAL UND FULL SPECTRUM IMAGING (HSI)

Ein weiterer optischer Sensortyp sind Hyper- bzw. Full Spectral Imagers. Die Bezeichnungen „hyper" und „full" sind einer semantischen Unschärfe zu schulden, da in der Literatur mal von „hyperspectral" und mal von „fullspectral" gesprochen wird. Im Kontext dieses Reports wird von „hyperspectral imagers" (HSI)

die Rede sein, da kein Sensor genuin das *gesamte* Lichtspektrum abdecken kann. Nichtsdestotrotz zeichnen sich HSIs dadurch aus, einen weit größeren Teil des Spektrums abzudecken als reguläre oder Infrarotkameras, wodurch sie in der Lage sind, verschiedene Materialien einzig anhand ihrer Lichtreflexionen zu erkennen (Manolakis und Shaw 2002: 30).

HSIs produzieren volumetrische „Bild-Würfel", d.h. mehrere überlappende Bilder – eines pro Spektralband. Manolakis und Shaw berichten von bis zu 144 unterschiedlichen Bändern (2002: 31). Insgesamt können HSIs bis zu 388 Millionen Datenpunkte pro Bild erzeugen (Adão et al. 2017: 7), was zwar weit geringer ist als die Auflösung, welche „Gorgon Stare" produzieren kann, nichtsdestotrotz aber wesentlich mehr Informationen enthält. Die Daten, welche von HSIs erzeugt werden, können nämlich direkt von Algorithmen erfasst werden, während im Falle von Gorgon Stare zunächst mindestens ein Klassifizierer Objekte von Interesse erkennen muss.

Ein Algorithmus kann mit HSI-Daten arbeiten, indem er die verschiedenen Reflexionen mit einer Datenbank abgleicht, in welcher bekannte Materialien (beispielsweise Stahl, Asphalt oder Gummi) abgespeichert sind (Manolakis und Shaw 2002: 30). Es geht aber auch umgekehrt, indem lediglich nach „Anomalien" in den Bilddaten gesucht wird: Beispielsweise könnte ein kleines metallisches Objekt umgeben von Holz, Gras und Sand ein Auto sein (Adão et al. 2017: 10).

Im Fall von HSIs sind militärische Anwendungen bereits bekannt: In Afghanistan wurden diese beispielsweise verwendet, um „tausende Kilogramm" (Knight 2019) Sprengstoff zu entfernen. Auch zum Erkennen von Landminen aus der Luft werden HSIs eingesetzt (Maathuis und Genderen 2004). Die Anforderungen, welche HSI-Algorithmen an die Hardware stellen, sind vergleichsweise gering, da das Auffinden von bekannten Materialien oder Anomalien durch einfaches, statistisches Kategorisieren erfolgt. Daher ist es denkbar, dass zumindest eine vorläufige Analyse der HSI-Daten auf der Drohne selbst stattfinden und Bilddaten selektiv an Command & Control übermittelt werden können.

4.4 LIGHT DETECTION AND RANGING (LIDAR)

Bei LiDAR-Sensoren handelt es sich um Abstandsmesser, welche aus der Reflexionszeit eines Laserstrahls von einer Oberfläche die Entfernung dieser Oberfläche zum Sensor berechnen. Es existiert für LiDAR-Sensoren auch das Synonym

LADAR (Laser Detection and Ranging), eine rein semantische Unterscheidung (Alonzo 2013). LiDAR-Sensoren funktionieren genau wie Abstandsmesser, die vielfach beim Handwerken Verwendung finden, nur, dass sie zusätzlich *sensor fusion* nutzen können, um nicht nur Entfernungen zu bestimmen, sondern vielmehr die exakte Position der gemessenen Punkte. Dazu ziehen sie Daten von Positionssensoren wie dem GPS oder „inertial navigation systems“ (INS) heran, vgl. dazu den folgenden Abschnitt 4.5.

Die Auflösung von LiDAR-Sensoren ist starken Schwankungen unterworfen: sie reicht von wenigen Zentimetern bis hin zu einem Meter pro Punkt (vgl. hierzu die Angaben bei Anderson, Thompson und Austin 2005: 3889; Bas et al. 2018: 4096; Qu et al. 2014: 1; Wallace, Lucieer und Watson 2014: 7619; Yin und Wang 2019: 36). Weiterhin spielt laut Bas et al. (2018: 4073) auch der Winkel, aus welchem gemessen wird, eine Rolle. Um höhere Objekte verlässlich in den Daten erkennen zu können, bietet sich eher ein niedriger Winkel an, da bei einem hohen Winkel das Objekt wie ein Fehlerpunkt aussieht. In allen Fällen allerdings ist eine Einschränkung von LiDAR-Sensoren, dass sie, weil sie auf Lichtreflexionen aufbauen, nicht durch Wolken hindurch funktionieren, da das Wasser dort die Lichtstrahlen zu sehr zerstreut – denn es handelt sich wie bei HSIs und Kameras um optische Sensoren.

LiDAR-Sensoren produzieren maschinenlesbare Daten, die von Algorithmen direkt genutzt werden können, um Höhenprofile vom überblickten Gelände erstellen zu können. Hier ist ein möglicher Anwendungsfall für das Militär die Generierung von 3D-Karten, die in Command & Control-Stützpunkten mittels *information integration* mit weiteren Informationen angereichert werden könnten, wie es beispielsweise im SIMDIS SDK (vgl. Kapitel 7) bereits vorgesehen ist. Ein weiterer Anwendungsfall ist die automatische Navigation von Fahrzeugen, obgleich sich hier, wie oben bereits angedeutet, ein Trend zur Verwendung von Kameradaten abzeichnet und LiDAR-Sensoren teuer sind (vgl. Crowe 2019).

4.5 INERTIAL NAVIGATION SYSTEM (INS) UND SATELLITENNAVIGATION

Flugzeuge und Wasserfahrzeuge verfügen über Positionssensoren. Zwei sind besonders prominent anzutreffen und sollen hier vorgestellt werden. Inertial Navigation Systems sind *relative* Positionsmesser. Sie benötigen zu Beginn der Messung einen absoluten Punkt, von dem aus in der Folge nur relative Ab-

weichungen durch Veränderungen im Luftdruck (Höhe) oder im umgebenden Magnetfeld bestimmt werden. Ihre Achillesferse ist der sogenannte „drift-bias error“: Kleine Messfehler potenzieren sich durch die relative Messung. Zwar gibt es Methoden, diesen internen Fehler zu minimieren (vgl. Cechowicz 2017; Ishibashi et al. 2007), doch weitaus gängiger ist die Verwendung des sogenannten Kálmán-Filters (Kálmán 1960) bzw. des nicht-linearen Kálmán-Filters (Fairfax und Fresconi 2012: 7–8). Bei diesem Algorithmus handelt es sich um einen Filter, welcher mithilfe anderer Sensordaten (z.B. GPS) den Messfehler ausgleichen kann.

Bei Satellitennavigationssystemen – wie dem US-amerikanischen Global Positioning System (GPS), dem europäischen Galileo, dem indischen IRNSS, dem russischen GLONASS und dem chinesischen BeiDou – handelt es sich um *absolute* Positionsmesser. Anders als bei INS bleibt bei Satellitennavigation der Messfehler also mehr oder minder konstant, da der Sensor die eigene Position immer wieder durch die Signale der Satelliten aktualisiert. Allerdings kann ein Satellitennavigations-Sensor sowohl gejammt werden, d.h. die Signale werden so sehr verzerrt, dass der Sensor die Position nicht mehr verlässlich bestimmen kann, oder er kann gespooft werden, d.h. es werden gefälschte Signale gesendet, welche die richtigen Daten überlagern und dem Sensor somit eine falsche Position vorgaukeln (vgl. für Spoofing-Methoden Tippenhauer et al. 2011).

INS- und Satellitennavigations-Sensoren können zusammen verwendet werden, wobei die Satellitennavigation von Zeit zu Zeit verwendet werden kann, um dem INS wieder die richtige Position zu geben. Interessant in dieser Hinsicht ist die Kaperung einer amerikanischen Drohne durch den Iran, was laut Berichten mittels GPS-Spoofing geschah (vgl. Rawnsley 2011). Dies legt nahe, dass entweder das Spoofing des iranischen Militärs sehr ausgeklügelt war, sodass auch ein Vergleich mit der Position des INS-Sensors keine auffällige Abweichung ergeben hat, oder dass die Drohne keinen Sicherheitsmechanismus besaß, welcher die beiden Signale auf Auffälligkeiten miteinander abgeglichen hat.

Positionssensoren können heute als sogenannte MEMS, d.h. mikro-elektromechanische Sensoren gebaut werden – kleiner als eine Fingerkuppe. Dies hat bereits dazu geführt, dass eine Forschergruppe eine Mörsergranate mit MEMS-Sensoren ausgestattet hat, um die ballistische Flugbahn mitten im Flug noch zusätzlich zu korrigieren (Fairfax und Fresconi 2012).

Positionssensoren könnten benutzt werden, um die Positionen aller Personen und Maschinen im Einsatzgebiet zu überwachen. Es ist denkbar, dass diese Daten in Command & Control-Stützpunkten mit topografischen Daten (Kartenmaterial) verbunden werden, um den Überblick über das Einsatzgebiet zu verbessern (vgl. auch hierfür das weiter unten vorgestellte SIMDIS SDK).

4.6 RADIO DETECTION AND RANGING (RADAR)

Radars gehören zu den ältesten Sensoren im militärischen Arsenal. Sie messen Position und Geschwindigkeit von Objekten, indem sie Radiowellen ausstrahlen und die Echos analysieren. Heute gibt es zwei grundsätzliche Typen von Radars: Sogenannte „Fixed Aperture"-Radars, zu erkennen an den rotierenden, aber festen, Antennen. Außerdem gibt es „Synthetic Aperture"-Radars. Diese bestehen ausschließlich aus nicht-beweglichen Teilen. Während bei „Fixed Aperture"-Radars die Antenne immer dorthin gerichtet werden muss, wo die Radiowellen hingelangen sollen, wird derselbe Effekt bei „Synthetic Aperture"-Radars durch Veränderung des Abstrahlwinkels („beam forming") erzeugt.

Eine wichtige Entwicklung in den letzten Jahren ist das sogenannte „Phased-MIMO"-Radar (Haimovich, Blum und Cimini 2008; Hassanien und Vorobyov 2010). MIMO steht für „Multiple Input, Multiple Output" und beschreibt letztlich ein Radar-System mit mehreren Sendern und mehreren Empfängern, das mit nur 33 Prozent der Signalqualität bessere Erkennungsraten als herkömmliche Radars erzielen kann (Haimovich, Blum und Cimini 2008: 123, Abb. 4). Dadurch ist denkbar, dass auch Tarnkappenflugzeuge bzw. Raketen besser erkannt werden können, da es bei mehreren Radarquellen schwieriger ist, diese effektiv zu zerstreuen.

Durch ihre Fähigkeit, Entfernung und Geschwindigkeit von Objekten zu messen, können Radar-Geräte für Kollisionserkennungssysteme verwendet werden (vgl. Fasano et al. 2015). Weitere Anwendungsbereiche umfassen die Aufklärung und die sogenannte *terminal illumination*[14] von durch Raketen anvisierten Zielen. Auch Nahbereichsverteidigungssysteme *(Close-In Weapon System, CIWS)* wie die US-amerikanische Phalanx oder die russische Kortik verwenden Radar, um Ziele zu identifizieren.

5 Eingebettete und integrierte Systeme

Nachdem nun ein Überblick über die verschiedenen Daten gegeben wurde, welche von Sensoren an verschiedenen Orten des jeweiligen Einsatzgebietes erfasst werden können, ist eine weitere Frage, ob die Daten bereits auf der Drohne ausgewertet werden können, wo sie gesammelt werden, oder die Rohdaten an andere Stellen weitergeleitet werden müssen. Ist ersteres möglich, so können Daten von Interesse selektiv gesendet und (aus Sicht des Algorithmus) uninteressante verworfen werden. Ist dies nicht möglich, erhöht sich die benötigte Bandbreite zur Übermittlung der Daten.

Bereits im letzten Kapitel wurde immer wieder indirekt auf die notwendige Rechenleistung von Algorithmen zur Datenauswertung Bezug genommen. Letztlich gibt es, wie im Kapitel zur Datenprozessierung noch einmal gesondert erklärt wird, zwei Gruppen, in die sich die militärisch relevanten Algorithmen einteilen lassen: Einmal relativ einfache Algorithmen, die auch auf leistungsschwächeren Systemen lauffähig sind, und komplexe Algorithmen in Form von modellbasierten Klassifizierern, welche durch ihre großen Modelle erheblich mehr Rechenleistung verlangen. Von daher ist eine weitere Kategorie, die im Kontext militärischer Software relevant ist, die Unterscheidung in eingebettete und integrierte Systeme.

Grundsätzlich ist die Grenze zwischen eingebetteten und integrierten Systemen schwammig, da fast alle eingebetteten Systeme auch als integriertes System definiert werden können. Der vorliegende Research Report nutzt daher folgende Arbeitsdefinition:

> Ein eingebettetes System („embedded system“) ist ein physisch abgeschlossenes System, welches autark, d.h. ohne externe Kabel mobil ist. Ein eingebettetes System im Kontext dieses Textes ist also über die Hardware und nicht über die Software definiert. Ein integriertes System wiederum kann in zwei Gruppen unterteilt werden: Erstens ein Zusammenschluss von eingebetteten Systemen, wie es beispielsweise im „System of Systems“-Ansatz des US-Militärs gesehen wird, d.h. bestehend aus Drohnen, Schiffen und Kampfjets. Zweitens, ein Zusammenschluss von Softwarekomponenten, die ihrerseits ganz ähnlich dem „System of Systems“-Ansatz durch die Interaktion leistungsstärker sind als die einzelnen Softwarekomponenten.

Beispiele für eingebettete Systems sind demnach die von Militärs mittlerweile ubiquitär eingesetzten Drohnen, mobile Radarsysteme oder ferngesteuerte Wasserfahrzeuge wie die „Sea Hunter" der US-Navy. Zur Gruppe der integrierten Systeme wiederum zählen unter anderem das AEGIS-System oder die SIMDIS SDK. Das AEGIS-System ist ein Kampfsystem für Marineschiffe und besteht aus Radar-Sensoren, die mittels Computer an die Waffen der Schiffe gekoppelt sind und das sowohl für Luftraumüberwachung als auch als Frühwarnsystem genutzt werden kann (vgl. Kimmel 2009). Das AEGIS-System wird bereits seit den 1970er Jahren verwendet. Die SIMDIS SDK wiederum ist eine Bibliothek von Software-Anwendungen, die genutzt werden kann, um automatisch verschiedenste Sensordaten grafisch darzustellen – es handelt sich also um ein System zur *information integration*. Sie wird seit den 1990er Jahren vom U.S. Naval Research Laboratory (2019: 5) beständig weiter entwickelt und steht als Open Source-Anwendung zum kostenlosen Download bereit.

Doch es stellt sich die Frage, *ab wann* das Zusammenschalten von mehreren Hard- und Softwarekomponenten zu einem integrierten System führt. Definitionen aus Wissenschaft und Militär sind hier sehr vage:

> Also, the integration of sensors, platforms, and command organizations, along with advances in computer processing, precise global positioning, and telecommunications, will provide the capability for determining accurate locations of friendly and enemy forces and for collecting, processing, and distributing relevant data to thousands of locations. (Manthorpe 1996: 310)

Eine weitere Definition, die sich stärker an der computerwissenschaftlichen Lesart von integrierten Systemen orientiert, das heißt auch den menschlichen Faktor mit einbezieht, da auch die bedienenden Personen streng genommen Teil des Systems sein können, lautet:

> Simply stated, a system is an integrated composite of *people*, *products*, and *processes* that provide a capability to satisfy a stated need or objective. (Defense Acquisition University 2001: 3, eigene Hervorhebung)

Dies zeigt die fließenden Übergänge zwischen einzelnen und miteinander verschalteten Systemen.

Zuletzt stellt die Integration hardwarebasierter Systeme militärische Operationen vor erhebliche Probleme. Wie Zachary Davis (2019: 10–11) berichtet, sind Systeme verschiedener Hersteller mit Blick auf Schnittstellen, Komponenten und ver-

wendete Protokolle bislang inkompatibel und der Versuch, diese miteinander zu koppeln, könnte durch komplexere Algorithmen in Zukunft gar noch erschwert werden. Kurzum heißt dies, dass die Synergien mehrerer sensorgestützer Systeme in verschiedenen Einsatzgebieten in Ermangelung effektiver *information integration* bislang offenbar nicht durch das Militär nutzbar sind.

Der maßgebliche Grund für diese Unterscheidung in eingebettete und integrierte Systeme ist, dass eingebettete Systeme mit Einschränkungen mit Bezug auf die Rechenleistung konfrontiert sind, die sich bei softwarebasierten integrierten Systemen nicht oder nur in geringerem Maße stellen. Während integrierte Systeme meist im Hintergrund über ganze Rechenzentren verfügen (mobile Rechenzentren im Einsatzgebiet befinden sich beispielsweise in Schiffscontainern, vgl. Bauer 2018), müssen eingebettete Systeme mit den Ressourcen auskommen, welche sich auf der Plattform befinden. Militärische Akteure werden wahrscheinlich darauf achten, dass die verbauten Komponenten keine allzu hohen Energiewerte aufweisen, damit möglichst lange Flugzeiten erzielt werden können. In der Industrie wird meist von sogenannten „SWaP"-Restriktionen gesprochen, das heißt „Size, Weight, and Power" – Größe, Gewicht und Energie(verbrauch) (vgl. ADLink 2019). Zudem sind eingebettete Systeme meist weniger modular als reguläre Server und Computer: Zumeist handelt es sich um aufeinander abgestimmte Komponenten, weshalb die Auswahl der Komponenten wichtig ist (Davis 2019: 10–11).

Während Arbeitsspeicher und in steigendem Maße Festplatten weniger problematisch in ausreichender Menge auf solchen Systemen verbaut werden können, sind vor allem die eigentlichen Recheneinheiten diejenigen, welche erheblich mehr Energie verbrauchen.[15] Daher, und insbesondere mit Blick darauf, dass modellbasierte Klassifizierer idealerweise *mehrere* Prozessorkerne benötigen, ist davon auszugehen, dass die Auswahl, welche Prozessorkerne auf den Systemen zum Einsatz kommen, eine relevante Frage im technischen Design einer Drohne ist. Hinzu kommt, dass verschiedene Arten von Recheneinheiten besser für bestimmte Arten von Aufgaben geeignet sind.

Es gibt heute drei große Gruppen von Recheneinheiten: Central Processing Units (CPU) sind Allrounder und in allen Computern installiert. Graphical Processing Units (GPU) sind spezialisiert auf parallele Berechnungen und daher Teil von Grafikkarten. Tensor Processing Units (TPU) sind eine neue Entwicklung von Google (Sato, Young, und Patterson 2017) und spezialisiert auf das schnelle Berechnen von sogenannten Tensoren, also Containern für große Matrixmulti-

plikationen, die vor allem beim Training von modellbasierten Klassifizierern eine Rolle spielen. Wie Wang, Wei und Brooks schreiben: „TPU is highly-optimized for large batches and CNNs, [...] GPU shows better flexibility and programmability for irregular computations, [... and] CPU has the best programmability, [...] and it supports the largest model because of large memory capacity“ (2019: 8).

Eingebettete Systeme gemäß der Definition dieses Reports finden im militärischen Bereich Verwendung für Aufklärung in Form von Drohnen, aber zunehmend auch zum Schutz von Soldat*innen in gefährlichen Situationen – so werden Bombenentschärfungen mehr und mehr durch Roboter vorgenommen und aktuell wird an immer kleineren Drohnen gearbeitet, welche die *situational awareness* der Soldat*innen im Feld steigern sollen. Zwei Trends werden die weitere Entwicklung von solchen Systemen vermutlich maßgeblich mitbestimmen und sollten daher Berücksichtigung für rüstungskontrollpolitische Ansätze finden: Einerseits die Miniaturisierung von Hardware-Elementen, sodass in der Tendenz mehr Rechenleistung in kleinere Systeme verbaut werden kann. Andererseits aber – und dies wird im Kapitel zur Datenprozessierung noch angesprochen – auch der Trend hin zu immer schnelleren und ressourcensparenden Klassifizierern, sodass wir in Zukunft davon ausgehen können, dass solche Systeme wahrscheinlich mehr und mehr Daten direkt auf der eigenen Hardware vorverarbeiten können.

6 Kommunikation

Unabhängig davon, ob die mittels Sensoren gesammelten Daten bereits auf der Drohne vorverarbeitet werden konnten oder nicht: Sie müssen zu Command & Control gesendet werden, um dort in die Entscheidungsfindung einfließen zu können. Dies geschieht heute über weite Strecken mittels kommerzieller Internetverbindungen. Doch die erste Wegstrecke müssen die Daten über Funk nehmen. Die Bandbreiten und Störanfälligkeiten militärischer Datenlinks im Einsatzgebiet sind Fokus dieses Kapitels.

Für mobile Anwendungen sind zwei Kommunikationswege relevant. Erstens sogenannte Line-of-Sight (LoS)-Kommunikation, d.h. die Übertragung von Infor-

mationen auf direktem Weg zwischen Sender und Empfänger und zweitens Satellitenkommunikation, welche auch „over the horizon“ (OTH), d.h. ohne direkte Verbindung funktioniert.

6.1 LINE OF SIGHT-KOMMUNIKATION: TACTICAL DATA LINKS (TDL)

Weltweit gibt es eine Vielzahl militärischer Datenverbindungen.[16] Die aktuell von der NATO verwendeten Taktischen Datenlinks (TDL) heißen „Link 11“ (wird derzeit durch neuere Versionen ersetzt), „Link 16“ (aktuell verwendet) und „Link 22“ (zukünftig, aber teilweise bereits umgesetzt, vor allem in Bezug zum F-35-Projekt). Weitere Datenlinks sind entweder nie in Betrieb genommen worden, oder überholt (Link 4 beispielsweise wurde in den 1950er Jahren verwendet).

Link 11 und Link 16, die derzeit maßgeblichen Datenlinks, sind untereinander inkompatibel, was dazu führt, dass ältere AEGIS-Systeme nicht mit F-16-Jets kommunizieren können, da erstere nur über Schnittstellen zu Link 4, 11 und 14 verfügen, neuere Versionen der F-16 allerdings Link 16 verwenden (vgl. Kimmel 2009: 31, Abb. 2). Dies war unter anderem ein Grund für die Entwicklung von Link 22, welcher Link 11 langfristig ersetzen soll und als „Brücke“ zwischen Link 11 und Link 16 fungiert (vgl. Longdon 2013 für einen Überblick über Link 22).

Abgesehen von Kompatibilitätsproblemen stellen sich aber noch drei weitere Probleme, von welchen zwei mit Bezug auf Satellitennavigations-Sensoren in Kapitel 2 bereits angesprochen wurden: Erstens Spoofing und zweitens Jamming der Signale (vgl. Tippenhauer et al. 2011). Das dritte Problem spezifisch bei Datenlinks ist allerdings die Datendurchsatzrate. Durch die hohen Sicherheitsanforderungen wie „frequency hopping“ (häufiges Wechseln der Frequenz zum Schutz gegen Jamming) und leistungsstarke Verschlüsselung liegt der effektive Datendurchsatz selbst im „enhanced mode“ für Link 16 bei knapp 150 Kilobyte pro Sekunde (Martinez-Ruiz et al. 2010: 1163). Zum Vergleich: Der aktuell verbreitetste Video-Codec H.264 (assoziiert mit dem MPEG-4-Videoformat) produziert bei einem akzeptablen Kompromiss zwischen Bildqualität und Datenmenge rund fünf Megabyte Daten pro Sekunde.[17]

Dass sich die geschützten militärischen Datenlinks nicht für die Übertragung von größeren Datenmengen eignen, wird im Interview vom früheren Air Force-Mitarbeiter Marty McDonough bestätigt: „Streaming video can also be transmitted

over Link 16 but eats up all available bandwidth“ (McDonough 2010). Allerdings sollte einschränkend erwähnt werden, dass eine jüngere Studie erstaunlich eindeutig die Frage nach der Bandbreite für Videos beantwortet hat: „1Mbps is enough“ (Schmitt et al. 2016) – ein Megabit (d.h. 125 Kilobyte) je Sekunde reiche also aus. Das entspricht in etwa der Datendurchsatzrate von Link 16.

Die Lösung, welche das Militär für große Datenmengen letztlich aber gefunden hat, ist, Daten aus dem Einsatzgebiet heraus mittels kommerzieller Satelliten zu übertragen.

6.2 OVER THE HORIZON: KOMMERZIELLE DATENLINKS UND DROHNENÜBERWACHUNG

Eine US-amerikanische Reaper-Drohne (MQ-9), wie sie aus Afghanistan bekannt ist, kann bis zu 4 Megabyte an Daten pro Sekunde über kommerzielles KU-Band-Radio senden (Erwin 2017), was bestätigte Praxis des US-Militärs ist und mutmaßlich auch einer der Gründe dafür war, dass über Jahre hinweg Al Qaeda und die Taliban mit gerade einmal 26 Dollar an Equipment die Videofeeds mitschneiden konnten (vgl. Cole 2009). Sofern die Daten nämlich nicht verschlüsselt sind, genügt eine Empfangsantenne und eine Dekodiersoftware, wie sie von Al Qaeda genutzt wurde.

Eine RAND-Studie von 2014 hat die Übertragungsraten verschiedener militärisch relevanter Kommunikationskanäle analysiert und gibt einen guten Überblick über die Bandbreite (Porche 2014). Während ein generischer Internetanschluss bis zu 5 Gigabyte pro Sekunde übertragen kann, kann ein Satellitenuplink nur rund 5 Megabyte pro Sekunde übertragen. Das bedeutet, dass ein Datensatz von einem Terabyte (1.024 Gigabyte) in ersterem Fall rund vier Minuten, in letzterem jedoch drei Tage zum Übertragen braucht (Porche 2014: 16, Tab. 3.1). Allerdings scheint es hier seit 2014 erhebliche Fortschritte gegeben zu haben. McLain und King berichteten nur drei Jahre später, dass es bereits in den 1990er Jahren Satelliten mit einer theoretischen Durchsatzrate von bis zu 10 Gigabyte pro Sekunde gegeben habe und dass das derzeit aktuelle SpaceX-Projekt „StarLink“ Datenraten von fast 3 Gigabyte pro Sekunde und Satellit und als Gesamtsystem über 12 Gigabyte pro Sekunde liefern könnte (2017: 7, vgl. auch 13, Tab. 1).

Insbesondere bei kommerziellen Kommunikationsanbietern ist das Militär natürlich nicht der einzige Kunde, weswegen man von etwas geringerem Datendurchsatz ausgehen kann – allerdings ist fraglich, wie groß dieser tatsächlich ist. In jedem Fall ist das Operieren von dutzenden Überwachungsdrohnen unter diesen Vorzeichen absolut möglich und hier scheinen Engpässe in der Zukunft tendenziell überwunden zu werden – zumindest, wenn es zu einem großflächigen globalen Kommunikationsnetz kommen sollte, wie Unternehmen wie SpaceX es planen.

Mit zunehmender Abdeckung des gesamten Planeten mit satellitengestütztem Internetzugang wird es zudem immer einfacher, mittels Drohnen jeden Flecken Erde zu überwachen, ohne dass sich ein Stützpunkt in der Nähe befinden muss. Aus rüstungskontrollpolitischer Sicht könnte dies problematisch werden, da mit zunehmender Dezentralisierung der Datengewinnung eine Verifikation der Einhaltung von Rüstungskontrollmaßnahmen schwieriger wird. Ob dies in Zukunft zu geringerer Nutzung taktischer Datenlinks für Drohnenüberwachung führen wird, ist allerdings allein auf Grundlage des derzeitigen Wissensstandes nicht einzuschätzen.

7 Datenprozessierung und -analyse

Nach dem Erzeugen der Daten auf Sensorplattformen wie Drohnen und ihrem anschließenden Transfer zu Command & Control-Infrastrukturen müssen die Daten noch analysiert werden, da der Zweck, zu welchem die Daten in erster Linie gesammelt werden, die Verbesserung militärischer Entscheidungsfindung ist. Rohdaten, kurzgefasst, sind zwar immer notwendige Grundlage von Datenanalyse, sprechen aber niemals „für sich“.

Jegliche Analyse erfolgt dabei über Algorithmen. Ein Algorithmus ist eine einfache Abfolge von Computerbefehlen, die in Reihe ausgeführt werden, um von einer Eingabe (Input) zu einer Ausgabe (Output) zu gelangen. Im Kontext dieses Research Reports relevant sind zwei Attribute von Algorithmen: Einmal die Komplexität des zugrundeliegenden Problems, und andererseits die Laufzeit.

Algorithmische Komplexität wird mathematisch mit NP (nichtdeterministisch polynomielle Zeit) beziffert. Als NP-vollständig werden Probleme bezeichnet, deren Komplexität mit wachsender Problemgröße (*n*) so stark steigt, dass ein Algorithmus bereits bei kleinen *n* unverhältnismäßig lange zur Lösung benötigt und damit unpraktikabel wird. NP-vollständig ist beispielsweise das sogenannte „Graph Shaping", welches von Wissenschaftlern der United States Military Academy als Ansatz zur Terrorismusbekämpfung vorgeschlagen wurde (Callahan et al. 2012). Zahlreiche Such-Probleme (darunter auch Bilderkennung) sind NP-vollständig, weshalb hier oftmals auf sogenannte *Heuristiken* zurückgegriffen wird, d.h. vorgefertigte Annahmen über den *vermutlich* besten Lösungsweg. Dies reduziert die benötigte Laufzeit der Algorithmen, da unwahrscheinliche Fälle von vornherein ausgeschlossen werden.

Die Laufzeit von Algorithmen wiederum bemisst die Dauer, welche nötig ist, um ein Problem zu lösen. Die Laufzeit wird meist in der O-Notation, z.B. *O(n)*, angegeben. Die O-Notation wird wie folgt gelesen: „Der Algorithmus benötigt auch im schlimmsten Fall („worst case") nicht länger als *n* Rechenoperationen". Viele Algorithmen werden daher optimiert, um eine Laufzeit von *log(n)* zu erreichen, da die Laufzeit mit steigendem *n* im Falle einer logarithmischen Laufzeit immer langsamer steigt. Ineffiziente Algorithmen weisen dagegen eher eine Laufzeit von 2^n auf, d.h. im schlimmsten Fall kommt der Algorithmus erst nach erheblicher Laufzeit zum Ende – wenn überhaupt.

Viele der nachfolgend zitierten Paper nutzen allerdings nicht die O-Notation, sondern geben die Laufzeit in Zeiteinheiten an. Das macht die Vergleichbarkeit der Effizienz beinahe unmöglich, da die zum Testen verwendeten Computer nicht standardisiert waren und der jeweils verwendete Code nicht veröffentlicht wurde.

7.1 ZIELERFASSUNGSALGORITHMEN

Zielerfassungsalgorithmen haben zum Ziel, im Kamera-Sichtfeld (FOV) potenzielle Ziele zu finden. Vor allem Raketen, ausgestattet mit Infrarotsensoren, werden mit dieser Technologie in Verbindung gebracht, aber auch zum Erkennen beweglicher Ziele auf Kamerabildern können vergleichsweise simple Algorithmen genutzt werden.

Die Annahme, dass sich Ziele bewegen, stößt allerdings an ihre Grenzen, wenn es um Parallax-Effekte geht. Denn wenn sich die Kamera bewegt, z.B. mit der

Drohne, erscheinen hohe Objekte wie Bäume oder Gebäude auch als bewegt, da sich ihre Spitze entgegen der Richtung der Kamera zu bewegen scheint (Shi et al. 2012: 2513). Um dem Problem des Erkennens von Bäumen als Ziele zu entgegnen, werden also weitere Annahmen benötigt. Ein Algorithmus berechnet beispielsweise die Position des Straßennetzes auf Bildern, und verwirft alle erkannten Objekte außerhalb dessen als Fehler (ebd.: 2512). Aufgrund der Vielzahl an Umgebungen, in welchen militärische Operationen durchgeführt werden können und der Tatsache, dass diese Vorannahmen Änderungen am Algorithmus erfordern, ist es unwahrscheinlich, dass militärische Software in diesem Fall auf solche Heuristiken zurückgreift.

Hauptsächlich werden zur Zielerfassung allerdings keine Kameras, sondern Infrarotsensoren verwendet. Insbesondere Lenkraketen setzen auf FLIR-Sensoren. Neuere Algorithmen wie der MSDU von Cao et al. (2015: 5) verfügen über eine Art „Gedächtnis“, da sie sich merken, wie ihr Ziel aussieht und ein Modell dessen im Speicher behalten. Dies hat einen anderen Algorithmus dazu befähigt, ein Flugzeug von Infrarot-Täuschkörpern („flares“) zu unterscheiden, während ältere Algorithmen erfolgreich getäuscht wurden und ihr Ziel verloren haben (Wu et al. 2019: 12). Beide Algorithmen funktionieren dabei in Echtzeit, da sie die Infrarotbilder mit 50 (Cao et al. 2015: 8–9) bzw. 23 (Wu et al. 2019: 15) Bildern pro Sekunde auswerten können.

Hyperspektrale Bilder haben im Gegensatz zu Kamerabildern den Vorteil, dass die enthaltenen Informationen direkt maschinenlesbar sind, da die Reflexions-Spektrogramme durch Algorithmen direkt ausgewertet werden können (Adão et al. 2017). Dabei gehen Algorithmen statistisch vor und klassifizieren Materialien anhand der Ähnlichkeit zu unter Laborbedingungen entstandenen Spektra (Manolakis und Shaw 2002: 29). Hierbei wird deutlich, dass HSI-Algorithmen ein wenig wie modellbasierte Klassifizierer arbeiten, nur sind sie wesentlich simpler, da sie das Spektrum verschiedener Materialien nicht *lernen* müssen. Selbst für „Anomalie-Detektion“, d.h. das Herausfiltern von wenigen distinkten Materialien (beispielsweise ein kleines Auto auf einem Feld) benötigen die Algorithmen keine Lernphase (Adão et al. 2017: 10).

Neuere Entwicklungen von HSI-Algorithmen – insbesondere im Jahr 2017 gab es zahlreiche Publikationen – weisen auf einen Trend, dass nicht nur die Materialien selbst („was?“), sondern auch ihre Position im Bild („wo?“) in Betracht gezogen werden (vgl. für einen Überblick Zhao, Du, und Zhang 2017: 1). Dadurch können neuere Algorithmen rund 80 Prozent der Ziele in Testbildern korrekt er-

kennen (Kang et al. 2017: 8, Tab. II und Abb. 8). Ein Problem für viele Algorithmen ist allerdings, dass diese einen Parameter brauchen, der angibt, wie groß die gewünschten Ziele in etwa sind (Kang et al. 2017: 7). Ist dieser Parameter nicht optimal, werden entweder zu viele Objekte als Ziele erkannt, oder zu wenige (Kang et al. 2017: 10). Viele HSI-Algorithmen sind allerdings vergleichsweise schnell und analysieren ein Bild in teilweise nur 270 Millisekunden (ebd.: 8–9), weswegen Algorithmen auch verschiedene Größen dieses Parameters automatisch durchtesten können (Taghipour und Ghassemian 2017: 5).

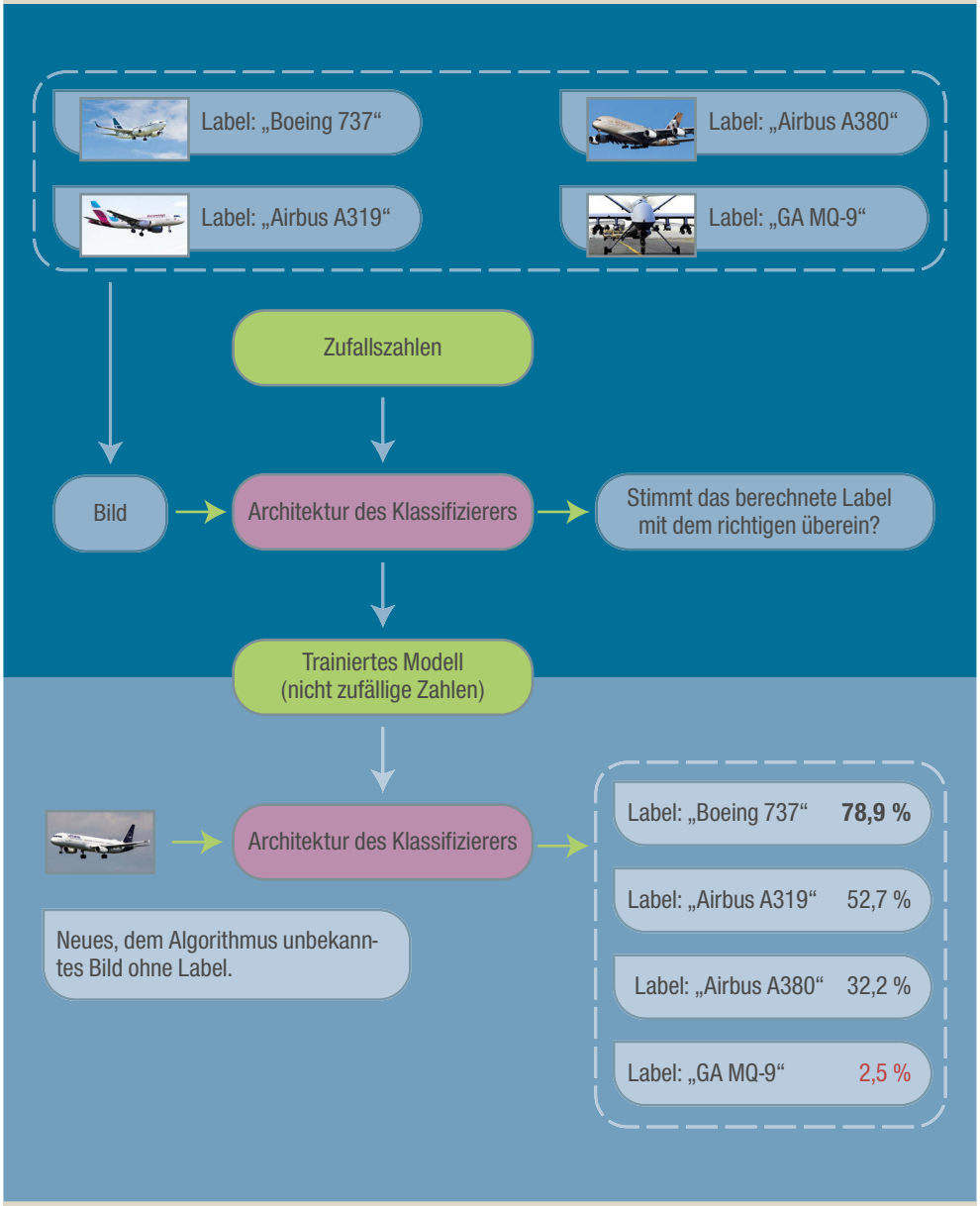
Aufgrund der teils sehr spezifischen Anforderungen einzelner Einsatzszenarien an Zielerfassungsalgorithmen ist ein schnelles Prototyping, d.h. das schnelle Entwickeln und Austesten eines Prototyp-Algorithmus, für das Militär wichtig. Zu diesem Zwecke gibt es ein quelloffenes Framework auf GitHub[18] (Crouse 2017), welches gängige Funktionen und Algorithmen bereitstellt, die in vielen Zielerfassungsalgorithmen verwendet werden, um die Entwicklung zu beschleunigen. Wie Crouse erklärt, enthält die Bibliothek „basic routines for filtering, coordinate conversions, assignment algorithms, physical effects, mathematics, and more“ (2017: 18). Dies ist ein prominentes Beispiel für die Intention hinter der *Code.mil*-Initiative des US Militärs.

7.2 MODELLBASIERTE KLASSIFIZIERER („NEURONALE NETZWERKE“)

Alle oben genannten Algorithmen haben gemein, dass sie simpel und leistungstechnisch effizient sind, aber sie arbeiten mit vielen Annahmen und Heuristiken. Infrarotalgorithmen können davon ausgehen, dass Ziele immer mehr Wärmestrahlung als ihre Umgebung abgeben und HSI-Algorithmen können davon ausgehen, dass Anomalien respektive spezifische Materialien Ziele darstellen. Alle diese Algorithmen aber stoßen an ihre Grenzen, wenn es darum geht, zu spezifizieren, *was* denn die Ziele sind. Hierfür wurden spezielle Algorithmen entwickelt, welche nach dem Prinzip funktionieren, ein Bild gemäß einem angelernten Modell in verschiedene Kategorien zu klassifizieren. Diese Algorithmen, die als „neuronale Netzwerke“ bekannt sind und im Kontext dieses Reports als Klassifizierer bezeichnet werden (vgl. die obigen Ausführungen zu technischen Begriffen), werden in immer mehr Anwendungen verwendet und daher ist stark davon auszugehen, dass sie in Zukunft auch in immer mehr militärischen Bereichen adaptiert werden (bzw. es bereits werden, vgl. Davis 2019). Dieser Dual-Use-Charakter stellt für eine potenzielle Rüstungskontrolle ein Problem dar.

Abbildung 2: Prozess des „Machine Learning“

Training: Für das Training benötigt der Klassifizierer drei Dinge: Eine Matrix aus Zufallszahlen, Trainingsbilder und die zugehörigen Labels. Während des Trainings werden dann die Zufallszahlen solange angepasst, bis der Algorithmus mithilfe der Zahlen die richtigen Labels der Trainingsbilder ausgibt.



Anwendung:
Im Feld werden nicht vorher beschriftete Daten (ohne Label) durch den Algorithmus kategorisiert. Der Algorithmus gibt mehrere Kategorien aus und weist ihnen Wahrscheinlichkeiten zu.

Quelle: eigene Darstellung.

Solche Klassifizierer sind eng mit dem Begriff des „machine learning“ verknüpft. Als „machine learning“ wird in der Informatik ein Prozess bezeichnet, bei welchem ein Algorithmus nicht für ein bestimmtes Problem implementiert wird, sondern für eine ganze Klasse[19] von Problemen. Solche Algorithmen folgen einer bestimmten Struktur, der sogenannten „Architektur“[20] und nutzen ein „Modell“, welches während der Phase des Lernens solange angepasst wird, bis der Algorithmus mithilfe dieses Modells die Eingabedaten korrekt zuordnen kann. Dieses Modell ist eine Matrix mit sehr vielen (zwischen 5 und 100 Millionen, vgl. Sato, Young und Patterson 2017) zunächst zufälligen Zahlen – den sogenannten Gewichten –, die durch das Training angepasst werden (vgl. hierzu Abb. 2). Vielfach wird davon im Bereich der Bilderkennung Gebrauch gemacht, wobei während der Lern-Phase Bilder durch den Algorithmus analysiert werden, und die einzelnen Gewichte des Modells solange auf Grundlage statistischer Wahrscheinlichkeiten verändert werden, bis die vom Algorithmus ausgegebenen Kategorien der Bilder oft genug korrekt sind. Der mathematische Begriff hierfür lautet „Gradientenverfahren“.

Ein Klassifizierer arbeitet anders als die oben angesprochenen Algorithmen. Während deren Output in einem exakten Wert besteht, handelt es sich beim Output eines Klassifizierers um eine Liste mit Wahrscheinlichkeiten. Und zwar gibt kein Klassifizierer nur eine einzige Kategorie, z.B. „Kampfflugzeug“ aus, sondern eine Reihe verschiedener Kategorien mit zugehörigen Wahrscheinlichkeiten, die angeben, wie sicher sich der Algorithmus ist, dass es sich um die genannte Kategorie handelt. Das erklärt auch, weswegen im vorangegangenen Absatz von „oft genug korrekt“ gesprochen wurde – es gibt noch keine standardisierte Methode, die Genauigkeit von solchen Klassifizierern anzugeben. Was sich allerdings durchgesetzt hat, ist die Angabe sogenannter „Top“-Wahrscheinlichkeiten. Ein Beispiel: Wenn die Genauigkeit eines Klassifizierers mit „93 % Top-5“ angegeben wird, heißt dies, dass sich in 93 von 100 Fällen die korrekte Kategorie eines Eingabe-Bildes unter den wahrscheinlichsten fünf Kategorien befindet.

Hier wird eine ganz zentrale sicherheitspolitische Implikation von solchen Klassifizierern deutlich, die derzeit im Diskurs kaum aufgegriffen wird: Sofern nämlich der Output solcher Klassifizierer direkt an andere Algorithmen weiter gegeben wird, ist die Wahrscheinlichkeit sehr hoch, dass die falsche Kategorie genutzt wird – ein Algorithmus hat nämlich nicht die Möglichkeit, gegenzuprüfen, dass es sich bei der Top-Wahrscheinlichkeit auch tatsächlich um die richtige Kategorie handelt. Um das „93 % Top-5“-Beispiel wieder aufzugreifen: Im Umkehrschluss heißt dies, dass in sieben Prozent aller Fälle die korrekte Kategorie gar

nicht in den oberen fünf Kategorien liegt und in den restlichen 93 Prozent liegt die richtige Kategorie entweder auf Platz 1, 2, 3, 4 oder 5 – und kein Algorithmus kann sicher sagen, welche nun die richtige ist.

Jeder Output eines Klassifizierers sollte also durch eine Person gegengeprüft werden, da es anderenfalls zu dramatischen Fehleinschätzungen kommen kann, die erstaunlich ähnlich zu jenem Zwischenfall 1988 sind, bei welchem eine iranische Passagiermaschine im persischen Golf durch das AEGIS-System der *USS Vincennes* fehlerhaft erkannt wurde und daraufhin abgeschossen wurde (Boulanin und Verbruggen 2017: 40). Auch wenn viele Details des Zwischenfalls unklar sind, scheint einer der Gründe für den Abschuss zu sein, dass das AEGIS-Kampfsystem den Airbus A300 fälschlicherweise als eine iranische F-14 Tomcat identifiziert habe (vgl. Lendon 2020).

In den vergangenen zwanzig Jahren sind diese Klassifizierer unter dem Namen „neuonaler Netzwerke“ immer größer geworden und arbeiten teilweise mit mehreren Millionen einzelner Gewichte. Das hat dazu geführt, dass selbst mit moderner Technik eine Echtzeit-Auswertung von Bildern kaum möglich ist. Allerdings gibt es seit Kurzem einen Trend in der Forschung, solche Klassifizierer wieder zu schrumpfen und zu versuchen, Bilderkennung mit der Effizienz kleinerer Algorithmen zu verbinden.

Ein Beispiel dafür ist das an der ETH Zürich entwickelte DroNet; ein kleiner Klassifizierer, welcher eine handelsübliche Drohne mit der Rechenleistung eines normalen Office-Laptops durch den Straßenverkehr leiten kann (Loquercio et al. 2018). Dadurch konnte die Drohne wesentlich längere Strecken zurücklegen (bis zu 245 Meter statt 75 Meter von konkurrierenden Algorithmen, vgl. ebd.: 6, Tab. II). Ein anderes Beispiel ist der YOLOv2-Klassifizierer („You Only Look Once“), dessen Auszeichnungsmerkmal ist, dass es sich zwar um ein klassisches „neuronales Netzwerk“ handelt, welches aber einerseits auf Smartphones in Echtzeit lauffähig ist und andererseits rund 8.000 Objekte mit einer geringen Fehlerrate erkennen kann (Redmon und Farhadi 2016). Ein zweiter Trend in der aktuellen Forschungslandschaft ist, Klassifizierer entweder durch Entfernen von einzelnen Knotenpunkten schneller und leichter zu machen, ohne dabei die Fehlerrate zu stark steigen zu lassen, oder aber die Netzwerke auf spezialisierte Hardware anzupassen (Venieris, Kouris, und Bouganis 2018). Letzteres wird sich aber wahrscheinlich nicht durchsetzen, da die spezialisierte Hardware teuer ist und die oben genannten Ansätze eine wesentlich einfachere Effizienzsteigerung versprechen.

7.3 ANGRIFFE AUF KLASSIFIZIERER MITTELS „ADVERSARIAL IMAGES“

Doch bei aller Fähigkeit von solchen Klassifizierern, Objekte in Bildern erkennen zu können, darf nicht vergessen werden, dass es sich letztlich um statistische Berechnungen handelt. Solche Klassifizierer sind sehr anfällig für Täuschungen mittels speziell präparierter Bilder. Forscher des Google Brain Teams haben beispielsweise erfolgreich mit einem Klassifizierer gerechnet – die Fähigkeit, bestimmte Objekte zu klassifizieren, wurde also missbraucht, um das Modell für nicht intendierte Zwecke zu nutzen (Elsayed, Goodfellow und Sohl-Dickstein 2019). Eine andere Attacke geht sogar noch weiter: Selbst ohne Wissen um die verwendete Architektur eines Klassifizierers konnten Forscher mit hohen Wahrscheinlichkeiten von teilweise über 90 Prozent die Architektur und sogenannte Hyper-Parameter (das heißt die Einstellungswerte) der Klassifizierer richtig angeben (Oh et al. 2018).

Die Gefahr solcher Angriffe ist im militärischen Bezug vielfältig. So kann eine Drohne, die mit Bilderkennung ausgestattet ist, beispielsweise dahingehend getäuscht werden, dass eine Flugabwehrstellung nicht als solche erkannt wird. Wenn solche Klassifizierer also auf Drohnen eingesetzt werden und nur die erkannten Ziele an Command & Control weitergeleitet werden, wären auch die befehlshabenden Offiziere de facto blind, trotz aller Kameras im Einsatzgebiet (siehe hierzu bereits Davis 2019). Dies kann zu Fehlentscheidungen führen, die im schlimmsten Fall Menschenleben kosten, falls auf Grundlage der falschen Informationen beispielsweise Angriffe befohlen werden.

7.4 TRAININGS-DATENSETS, ANWENDUNGSFÄLLE UND TRENDS

Jede Software benötigt Algorithmen, daher ist mit Bezug auf militärische Anwendungsfälle wichtig, zu erörtern, welche der beiden oben genannten Algorithmen für welche Problemfälle wichtig wird. Einfache und simple Algorithmen sind weitaus häufiger anzutreffen, da sie einfacher zu programmieren sind und wesentlich weniger Leistung benötigen als modellbasierte Klassifizierer. Letztere werden bislang vom Militär vermutlich seltener verwendet, was einerseits an ihrer relativen Neuheit im Vergleich zu simplen Algorithmen liegt, aber auch an den fehlenden Trainingsdaten. Wie in Abb. 2 gezeigt, werden händisch beschriftete Trainingsdaten benötigt, mit welchen Klassifizierer trainiert werden können. Doch das bedeutet, dass die zugrundeliegenden Fotos erst beschafft werden müssen.

Kommerzielle Klassifizierer haben es hier leichter, da mit dem ImageNet (vgl. Deng et al. 2009) seit einigen Jahren ein sehr ausführliches Datenset mit Fotos existiert – allerdings nur mit zivilen Objekten wie Haustieren oder Fahrzeugen. Ein Klassifizierer benötigt eine erhebliche Menge an Bildern, um ein Modell mit einer akzeptablen Trefferquote erstellen zu können. Bislang ist die Existenz keines solchen Datensatzes mit explizit militärischen Kategorien bekannt. Doch dieser wäre vonnöten, um solche Klassifizierer beispielsweise zur Überwachung eines weitläufigen Einsatzgebietes effektiv einsetzen zu können. Solange es solche Modelle nicht gibt, werden die großflächigen WAMI-Bilder von menschlichen Analyst*innen gesichtet. Es ist anzunehmen, dass Streitkräfte weltweit derzeit versuchen, an möglichst viel Bildmaterial zu gelangen, um selbst solche Modelle erstellen zu können.

Der erste Staat, dessen Armee es schafft, ein solches Modell zu trainieren, könnte einen Vorteil gegenüber anderen Streitkräften im Sinne der Reaktionsgeschwindigkeit erhalten. Diese Tatsache kann potenziell zu einem digitalen Wettrüsten führen (vgl. auch Davis 2019). Rüstungskontrollpolitisch relevant ist auch die Frage, inwieweit Daten aus den Aufklärungsflügen im Rahmen des Open Skies-Vertrages zum Training solcher Modelle genutzt werden könnten, da die Aufnahmeperspektive sehr ähnlich zu derjenigen von Drohnen ist. Doch abseits des Einsatzgebietes können Klassifizierer auch für andere Aufgaben verwendet werden, wie beispielsweise Logistik oder zur Erkennung von nur schwach sichtbaren Schäden an Ausrüstung.

7.5 DATENANALYSE IN DER CLOUD – MICROSOFT AZURE UND DER JEDI-VERTRAG DES US-VERTEIDIGUNGSMINISTERIUMS

Abschließend sei noch ein Trend angesprochen, der sich erst seit kurzem abzeichnet, mittelfristig aber in der Lage sein könnte, Datenprozessierung und die Speicherung von Daten nochmals fundamental zu verändern: Cloud Computing. Der Begriff „Cloud Computing“ ist dabei zuweilen irreführend, denn was gemeinhin als „Cloud“ bezeichnet wird, ist nichts anderes als Server-Infrastruktur.

Speziell interessiert eine Klasse von Produkten, die als „Functions as a Service“ (FaaS) bekannt ist. Es handelt sich dabei um einen Onlinedienst für Softwareanwendungen (vgl. für einen Überblick über verschiedene FaaS-Plattformen Yussupov et al. 2020): Ein*e Nutzer*in schreibt einen Algorithmus und lädt den

Quellcode dessen auf ein solches Cloud Computing-Portal hoch – beispielsweise Amazon Web Services (AWS), Google Cloud oder Microsoft Azure. Nun kann der*die Nutzer*in bei Bedarf („on demand") diesen Algorithmus von einem anderen Computer aus starten (beispielsweise über einen Link) und die Inputs mitliefern. Der Cloud Computing-Service nun ruft den Algorithmus auf, übergibt ihm die Eingabewerte und liefert die Ausgabe des Algorithmus an den*die Nutzer*in zurück.

Der oft beworbene Vorteil solcher Cloud Computing-Dienste ist, dass vor allem anspruchsvolle Algorithmen (das heißt vor allem Analysesoftware wie Klassifizierer) auf leistungsstarken Servern laufen können und nicht von der Hardware der Kund*innen abhängen. Ähnliches gilt auch für die Datenspeicherung auf solchen Cloud-Servern: Die Menge der Daten ist frei skalierbar, und so können beliebig große Datensets stets verfügbar gehalten werden.

Im Kontext dieses Reports ist wichtig, diesen Aspekt anzusprechen, da das Department of Defence erst Ende 2019 im Rahmen der „Joint Enterprise Defense Infrastructure" (JEDI) einen milliardenschweren Auftrag an Microsoft vergeben hat, um die Möglichkeiten solcher Cloud-Infrastruktur für militärische Zwecke auszuloten (vgl. Conger et al. 2019). Das Vergabeverfahren ist zwar derzeit aufgrund eines Verfahrensfehlers pausiert, doch letztlich ist es aus technischer Sicht unerheblich, ob Amazon oder Microsoft den Zuschlag bekommt, da sowohl die Amazon Web Services als auch Microsoft Azure auf denselben Prinzipien fußen. Wichtig wird, für welche Implementationsmethode sich das DoD letztlich entscheiden wird, denn es gibt zwei grundsätzliche Methoden, wie das Unternehmen eine solche Cloud-Struktur bereitstellen kann: Zum einen, indem es Server innerhalb von Gebäuden des US-Militärs einrichtet und mit derselben Software bespielt wie die kommerziell verfügbaren Server, oder ob das Militär völlig auf eigene Hardware verzichtet und diese beim Anbieter belässt. Auf der offiziellen Seite des DoD lässt sich die Beschreibung so lesen, dass auf die Infrastruktur bei Microsoft gesetzt werden soll: „Work performance will take place at *the awardee's place of performance*"[21] (eigene Hervorhebung).

Besonders in diesem Fall lohnt sich eine gesonderte Untersuchung, da dies bedeuten würde, dass auf ein und demselben Server Algorithmen von Privatanwender*innen und dem US-Militär laufen, was – je nach Sicherheitsstandard – zu einem erheblichen Sicherheitsproblem werden könnte. Denn so steigt die Gefahr, dass klassifizierte militärische Informationen an Dritte gelangen können. Genauer gesagt besteht hier die Gefahr, dass es zu einem neuen Wettrüsten

darüber kommt, welcher Staat als erstes Sicherheitslücken in der Infrastruktur solcher Anbieter entdeckt und für sich ausnutzen kann. Zudem weichen solche Initiativen die Trennung zwischen militärischer und ziviler Infrastruktur auf, weshalb zivile Cloud-Dienste, die auch militärisch genutzt werden, als militärisches Angriffsziel betrachtet werden könnten.

8 Datenspeicherung und -verwaltung

Nachdem Daten mittels Sensoren im Einsatzgebiet erhoben, verarbeitet und kommuniziert worden sind, müssen diese noch gespeichert werden. Dies dient einerseits dazu, potenzielle strategische und taktische Fehler im Nachhinein analysieren zu können, und andererseits ggfs. Klassifizierer weiter zu trainieren, um die Erkennungsraten zu steigern. Problematisch mit Bezug auf Daten sind vornehmlich die hohen Mengen, also „big data“. Eine Rand-Studie sprach bereits 2014 von einer „Datenflut“ (Porche 2014: 13), da bspw. die Navy-Drohne MQ-4C Triton pro Mission rund zehn bis zwanzig Terabyte an Daten generiere (vgl. ebd.: 14, Abb. 3.1). Um mit einer solchen Menge an Daten zurecht zu kommen, genügt es nicht, die Dateien wie gewohnt auf Computerlaufwerken abzulegen – es benötigt hierfür spezialisierte Systeme. Diese Sektion fokussiert sich weniger auf konkrete Software-Entwicklungen denn auf Paradigmen, die vornehmlich aus dem privaten Sektor kommen und versuchen, das Problem großer Datenmengen zu lösen.

8.1 VERTEILE DATEISYSTEME

Es gibt konzeptuell zwei Arten von „big data“, für deren Verarbeitung zwei unterschiedliche Lösungen entwickelt wurden, die im Prinzip ähnlich funktionieren, aber auf andere Algorithmen aufbauen. Einerseits müssen besonders große Dateien gespeichert werden. Andererseits müssen besonders viele Daten gespeichert werden. Es handelt sich also einerseits um wenige sehr große und andererseits viele sehr kleine Dateien, die unterschiedliche Probleme mit sich

bringen. Zunächst soll es um die Speicherung weniger sehr großer Dateien gehen, beispielsweise Videoaufnahmen von WAMI-Systemen wie „Gorgon Stare". Solche Videos, vor allem, wenn sie besonders lang sind, können schnell mehrere Terabyte Speicherplatz beanspruchen.

Für diesen Zweck wurden verteilte Dateisysteme entwickelt; zwei wichtige Vertreter sind das Hadoop Distributed File System (HDFS) sowie das Google Distributed File System (GDFS). Bei diesen handelt es sich um Software, die besonders große Dateien redundant (d.h. mehrere Duplikate) speichert und einen hohen Datendurchsatz aufweist. Dies geschieht vor allem dadurch, dass die Dateien gestreamed werden, d.h. sie werden nicht als Ganzes in den Arbeitsspeicher des Systems geladen, sondern Stück für Stück – ähnlich wie Filme bei Streaming-Anbietern. Während die Schreib- und Leseraten hoch sind und so große Datenmengen heute bereits gut verwaltet werden können, leiden alle diese Systeme unter dem sogenannten „small file problem". Damit wird der Effekt bezeichnet, dass die Schreib- und Leseraten bei kleinen Dateien dramatisch sinken und die Systeme im Endeffekt langsamer werden als normale Dateisysteme auf handelsüblichen Rechnern (Aizman, Maltby und Breuel 2019: 3). Ein möglicher Lösungsansatz besteht darin, kleine Dateien zusammenzufügen (ähnlich einem ZIP-Archiv), bis die resultierende Datei groß genug ist, um den Vorteil des Dateisystems wieder zu nutzen (Jiang, Li und Song 2010).

Große Dateien entstehen allerdings hauptsächlich mit Bezug auf Videodaten. Vor allem bei (vor-verarbeiteten) Sensordaten ist das Problem umgekehrt: Die einzelnen Datenpunkte sind extrem klein (nur wenige Byte), aber dafür sind sie wesentlich zahlreicher als Video-Aufnahmen. Daher werden solche kleinen Datenpunkte nicht als Dateien abgelegt, sondern in Datenbanken, was zum nächsten Punkt führt.

8.2 VERTEILTE DATENBANKEN

SQL bezeichnet eine Sprache für relationale Datenbanken, die ähnlich wie Excel-Dateien funktionieren, allerdings optimiert auf Geschwindigkeit und Integration mit anderen Softwarepaketen. Das Entwicklerteam hinter SQLite, einer SQL-Software, schreibt, dass für Dateien von rund 10 Kilobyte Größe der Datendurchsatz rund 35 Prozent höher sei als bei herkömmlichen Dateisystemen (vgl. SQLite o. J.). Daher werden vor allem für viele kleinere Datensätze heute Daten-

banken verwendet. Hierbei stellt sich allerdings ein anderes Problem: Weniger das effiziente Lesen und Schreiben von großen Dateien, sondern das Auffinden der einzelnen Einträge. Hierfür gibt es zwei maßgebliche Technologien, die in Kürze vorgestellt werden sollen.

„Sharding“ ist die englische Bezeichnung für Fragmentierung, das heißt das Aufteilen einer großen Menge von Datenpunkten auf mehrere Datenbanken (vgl. für einen Überblick über die Funktionsweise von Sharding Venkateswaran und Changder 2017; für einen implementations-agnostischen Algorithmus[22] Swart 2004). Dadurch wird das Speichern auf mehreren Servern wie auch das Scaling, d.h. das Installieren zusätzlicher Kapazitäten durch Hinzufügen neuer Server, vereinfacht. Aber dies führt zu einem neuen Problem: wie können diese Datenpunkte wieder aufgefunden werden?

Ein praktisches Beispiel ist die US-Gesichtserkennungssoftware Clearview, die im Januar 2020 im medialen Fokus stand (Hill 2020). Hinter der Anwendung befindet sich eine Datenbank mit drei Milliarden Bildern. Nehmen wir eine moderate Dateigröße der Bilder von 300 Kilobyte an, ergeben sich 900 Terabyte an Daten, also fast ein Petabyte, was unmöglich auf einem einzelnen Server zu speichern ist. Daher ist anzunehmen, dass Clearview Sharding betreibt. Das US-Unternehmen Instagram steht vor demselben Problem: Es muss Milliarden von Fotos speichern und gleichzeitig Nutzer*innen ermöglichen, diese Bilder binnen einer Sekunde abzurufen. Das Problem stellt sich ebenfalls für die Umsetzung des JEDI-Projektes des US-Militärs.

Während sich die Entwickler hinter Clearview mit öffentlichen Äußerungen zurückhalten, ist das Techniker-Team von Instagram offener in der Preisgabe der Information, wie genau der Prozess des Speicherns und Abrufens solcher Informationen abläuft (Instagram Engineering 2016). Jeder Datenbankeintrag benötigt eine einzigartige ID. Normalerweise ist die ID eine aufsteigende Zahl, wie bei Excel-Zeilennummern. Doch solange diese Zahl eindeutig ist, muss sie nicht zwangsläufig aufsteigend sein. Instagram nutzt diese Tatsache, um IDs zu generieren, deren genauer Wert völlig egal ist, solange sie bestimmte Voraussetzungen erfüllt. Die IDs, welche Instagram vergibt, lassen sich nämlich in ihrer binären Repräsentation (geschrieben als Reihe von Nullen und Einsen) in drei Teile aufteilen: Den Zeitpunkt ihrer Erstellung, die Nummer des Servers, auf welchem der Eintrag liegt und die eigentliche (aufsteigende) ID innerhalb der Datenbank. Nun ist es möglich, entsprechend des Datums nur jene Server nach einem Foto suchen zu lassen, auf welchen Fotos im fraglichen Zeitraum abgespeichert

wurden. Dies reduziert die Zugriffszeiten erheblich, da nicht sämtliche Server durchsucht werden müssen.

Bei Clearview dürfte ein ähnlicher Prozess im Hintergrund ablaufen: IDs werden vergeben, deren binäre Repräsentation als Nullen und Einsen sich in einzelne Bestandteile zerlegen lässt. Es ist denkbar, dass Clearview die Server entsprechend gewisser Kriterien wie Augenfarbe, Haarfarbe, Geschlecht oder Herkunft des Fotos aufgeteilt hat. Wenn nun die Gesichtserkennungssoftware von Clearview ein Bild analysiert hat, kann sie die mutmaßliche ID ähnlicher Fotos, mit welchen das erkannte Bild verglichen werden soll, beinahe vollständig rekonstruieren. Ein Beispiel: Eine Person hat braune Augen (Augenfarbe Nr. 3), schwarze Haare (Haarfarbe Nr. 12) und das Foto wurde in Los Angeles aufgenommen (Orts-Nummer 2.374), dann können die binären Repräsentationen dieser Werte (11, 1100 und 1001 01000110) aneinander gereiht werden, was im Beispiel 11 11001001 01000110 (als Dezimalzahl 248.134) ergäbe. Ähnliche Prozesse könnten in Zukunft militärische Datenspeicherung ähnlich responsiv gestalten. Insbesondere im Kontext des JEDI-Kontextes ist zu erwarten, dass das US-Militär von Microsoft eine Azure[23]-ähnliche Infrastruktur gestellt bekommt, für welche Techniken wie das Sharding von erheblichem Nutzen sein werden. Hier besteht ein sicherheitspolitisches Risiko, da durch die dezentrale Speicherung der Anreiz für Angriffe wächst, denn jede einzelne Datenbank des Systems stellt einen potenziellen Angriffspunkt dar.

8.3 DATENSPEICHERUNG MITTELS MAPREDUCE

Ein letzter Algorithmus zur Datenverwaltung, der hier vorgestellt werden soll, ist MapReduce, originär 2003 von Google entwickelt (Dean und Ghemawat 2008). MapReduce funktioniert, indem die zu verarbeiteten Daten als atomistisch aufgefasst werden, sodass sie in einem zweischrittigen Verfahren zunächst „gemappt“, d.h. Kategorien zugeordnet werden, bevor sie reduziert, d.h. platzsparend gespeichert werden. Dieses Verfahren lässt sich sowohl für verteilte Dateisysteme wie auch verteilte Datenbanken einsetzen.

Um wieder das Beispiel von Clearview zu nutzen: Hier ließe sich ein MapReduce-Algorithmus insofern einsetzen, als dass zunächst die gesammelten Fotos in einer Map-Funktion analysiert werden und gewisse Metadaten gesammelt werden, bevor in der Reduce-Funktion ähnliche Datensätze und ähnliche Fotos auf

gleichen Servern abgespeichert werden können, d.h. beispielsweise alle Fotos aus Los Angeles würden in einer Datei zusammengefasst auf einem Hadoop-Server abgelegt, während die zugehörigen Metadaten auf einem „Shard"-Server abgelegt würden. Wird nun ein Bild aus der LA-Region in der Datenbank abgefragt, benötigt das System aufgrund der oben angesprochenen Vorteile dieser Systeme nur zwei Server, anstelle potenziell tausender, abzufragen.

Alle drei hier vorgestellten Algorithmen sorgen im Prinzip dafür, die Laufzeit, d.h. $O(n)$, eines Datenverwaltungssystem in Richtung $log(n)$ anstelle 2^n zu bewegen. Abgesehen von dieser Effizienzsteigerung haben verteilte Datenbanken und Dateisysteme jedoch signifikante rüstungskontrollpolitische Implikationen. Beispielsweise wird hier die fehlende Lokalisierbarkeit von Software und Daten relevant. Je nach Umsetzung von Rüstungskontrolle könnten Streitkräfte versucht sein, ihre Daten so aufzuteilen, dass keine der einzelnen Datenbanken von Rüstungskontrollverträgen erfasst wird, sehr wohl allerdings die Verbindung der Daten aus diesen Datenbanken. Da diese Verbindung allerdings „on demand" geschehen kann, wird sie nur im flüchtigen (volatilen) Zwischenspeicher (RAM) der Server vorgehalten, was eine Verifikation von Regulationen unmöglich macht.

9 Fazit

Ziel dieses Research Reports ist eine Evaluation potenzieller militärischer Nutzung von Softwaretechnologien, um rüstungskontrollpolitische Folgen und Herausforderungen abschätzen zu können. Es hat sich gezeigt, dass Software andere Anforderungen an die Rüstungskontrollpolitik stellt, als die Nonproliferation konventioneller oder nuklearer Waffen. Angefangen von der fehlenden Materialität von Algorithmen bis hin zu den neuartigen Problemen, die verteilte Datenbanken für effektive Rüstungskontrolle darstellen können, ist ein tiefergehendes Verständnis der zugrundeliegenden Technologien notwendig. Dieser Report stellt vorläufige Ergebnisse dar und musste sich daher auf solche Software beschränken, welche für die militärische *situational awareness* genutzt werden kann. Im Folgenden werden die zentralen Erkenntnisse noch einmal gebündelt dargestellt und bewertet.

Mit Bezug auf die Sensorik lässt sich festhalten, dass die aktuellen Trends in die Richtung zeigen, vorhandene Sensoren qualitativ zu verbessern, also mit Bezug auf optische Sensoren eine höhere Auflösung und/oder eine höhere Präzision zu erreichen. Mit Bezug auf Navigationssensoren existieren Anstrengungen, diese gegen Störungen resistenter zu gestalten. Auch wird vereinzelt daran gearbeitet, mehrere Sensoren zu kombinieren (*sensor fusion*), um weitere Anwendungsspektren wie Kollisionsvermeidung, Flugbahnkorrektur von Projektilen oder auch eine höhere Radarauflösung zu ermöglichen.

Eingebettete Systeme wie beispielsweise Drohnen, so zeigt sich, werden in der absehbaren Zukunft vermutlich nicht an die Rechenleistung von festinstallierten Servern herankommen. Hier wird also wahrscheinlich bis dahin ein Zusammenspiel von mobilen Waffenplattformen, resilienten Kommunikationskanälen und leistungsstarken Rechenzentren maßgeblich sein. Allerdings wird aktiv daran gearbeitet, modellbasierte Klassifizierer soweit zu verkleinern, dass auch mit der reduzierten Leistung von Drohnen bereits Bilderkennung betrieben werden kann, womit dies als ein softwaretechnologischer Trend zu bezeichnen ist.

Kommerzielle Kommunikationskanäle wiederum verfügen über (noch) ausreichende Bandbreiten, um mehr als die aktuelle Auslastung des globalen Internet zu bedienen. Die Kommunikation im Einsatzgebiet ist durch die Mitigation von Gefahren wie Jamming und Spoofing nicht für die Übermittlung größerer Datenmengen geeignet. Kommerzielle Satellitensignale, die bereits vom Militär genutzt werden, bieten zwar höhere Bandbreiten, allerdings auf Kosten der Sicherheit. Inwieweit sich das Internet gegebenenfalls als Flaschenhals herausstellen könnte, kommt darauf an, ob die Entwicklung der Bandbreite des Internets mit dem Anstieg der durch militärische Einsätze generierten Datenmengen Schritt hält. Fraglich ist außerdem, welche Infrastruktur das US-Militär noch exklusiv vorhalten will, oder ob dort die Einschätzung überwiegt, für die Kommunikation vollständig auf kommerzielle Lösungen zu setzen.

Das zentrale Ergebnis aus dem Bereich Datenprozessierung und -analyse ist, dass es (wenngleich nicht trennscharf) zwei Gruppen von Algorithmen gibt – zum einen solche, die vergleichsweise einfach funktionieren und nicht sehr komplex sind. Diese können optimiert werden und sind meist auch auf mobilen Waffenplattformen mit ihren begrenzten Ressourcen lauffähig. Zum anderen gibt es die Gruppe der modellbasierten Klassifizierer, welche weit komplexer sind und erheblich höhere Rechenleistung beanspruchen, wie sie derzeit nur

von Servern geliefert werden kann. Es gibt hier aber bereits erste Ergebnisse, auch diese Klassifizierer soweit zu optimieren, dass sie auf mobiler Hardware lauffähig werden.

Mit Blick auf die Datenspeicherung und -verwaltung lässt sich konstatieren, dass es vor allem zwei zentrale Probleme gibt. Zum einen gibt es Datenbestände, bei welchen nur wenige, dafür sehr große Dateien abgespeichert werden müssen. Zum anderen bestehen viele Datensets aus sehr vielen, dafür sehr kleinen Dateien (bzw. besser: Einträgen). Mit Bezug auf das erste Problem gibt es bereits seit langem Implementationen wie HDFS oder GDFS, die weltweit bereits zu diesem Zweck verwendet werden. Bei sehr großen Datensätzen wiederum, die nur aus kleinen Einträgen bestehen, kommen verteilte Dateisysteme an ihre Grenzen, weswegen sich hier gerade neue Paradigmen herausbilden, etwa das „Sharding“. Besonders das US-Militär scheint an einer Adaption solcher neuen Paradigmen interessiert, weshalb (nach derzeitigem Stand) Microsoft über das JEDI-Projekt damit beauftragt wurde, Cloud-Infrastruktur für die zukünftig zu erwartenden, immer größeren Datenbestände zu liefern.

Durch die hohe Flexibilität von Softwaretechnologien lässt sich vermuten, dass außer den hier genannten Kategorien noch weitere Anwendungsfelder für Software möglich sind. Bilderkennungssoftware lässt sich beispielsweise nicht nur verwenden, um Ziele auf WAMI-Bildern auszumachen, sondern könnte auch für Schadenseinschätzungen verwendet werden – es müssten nur Modelle mit jeweils anderen Datensets trainiert werden. Insgesamt aber lässt sich sagen, dass die hier genannten Technologien aufgrund des spezifischen Fokus dieses Textes alle militärisch adaptiert werden können und es zum Großteil auch bereits werden. Fast alle derzeit relevanten Entwicklungen im Softwarebereich zielen dabei auf eine schnellere, breitere und präzisere *situational awareness* ab. Ebenfalls gibt es aber auch Bestrebungen, die Weitergabe von Befehlen über den umgekehrten Pfad (Steuerung bzw. Command & Control) mehr und mehr zu automatisieren und zu vereinfachen. Hierauf lag allerdings nicht der Fokus.

Der vorliegende Research Report identifiziert weiterhin mehrere kurz- und mittelfristige Trends in der Entwicklung von Software. Kurzfristig kann mit präziseren und effizienteren Algorithmen in der grundlegenden Vor-Verarbeitung von Daten auf den mobilen Waffenplattformen gerechnet werden, das heißt, dass die Lageeinschätzung im Einsatzgebiet auf immer schärfere und bessere Daten zurückgreifen kann. Auch die Proliferation von Sensoren in teils unübliche Be-

reiche (wie die Ausstattung einer Mörsergranate mit MEMS-Sensoren) sowie die weitere *sensor fusion* durch Kombination von beispielsweise Radar und optischer Kamera ist ein aktiv beforschtes Gebiet.

Mittelfristig ist allerdings noch viel offen. Für diesen Zeitraum zeichnet sich bereits ein Trend zu einer schrittweise Verbesserung der Effizienz von modellbasierten Klassifizierern ab. Insbesondere der durchaus wahrscheinliche Sprung solcher Klassifizierer auf mobile Waffenplattformen stellt eine rüstungskontrollpolitische Implikation dar, da dies die Ergebnisse der Klassifizierer weiter aus dem Blick von menschlichen Analyst*innen bewegt, sodass Kontrolle schwieriger werden könnte. Ein weiterer mittelfristiger Trend, dessen genaue Ausprägung allerdings noch nicht ersichtlich ist, wird sein, dass mehr und mehr auf Cloud-Computing-Lösungen gesetzt wird. Das bedeutet: Kleine Gruppen von Berechnungs-Aufgaben werden an einen entfernten Server geschickt, welcher die Ergebnisse dieser Berechnungen zurückgibt. Doch hier haben sich noch keine Paradigmen durchgesetzt – mit der Google Cloud, Amazon Web Services und Microsoft Azure gibt es drei große Anbieter von Cloud-Infrastrukturen. Doch die Software, die darauf genutzt wird, ist vielfältig und noch haben sich keine „best practices" herauskristallisiert, mit welchen sich diese Einschätzung untermauern ließe. Weiterhin werden die einzelnen Waffensysteme wahrscheinlich durch rechenstärkere Hardware und steigende Konnektivität immer verwundbarer für nicht-kinetische, digitale Angriffe. Da militärische Akteure darum bemüht sein werden, ihre eigenen, „smarten" Waffensysteme gegen Hackerangriffe abzusichern, könnte dies zu einer Art „digitalem Wettrüsten" führen, da ein erfolgreicher Angriff bedeuten kann, dass ein feindlich gesinnter Akteur effektiv die Kontrolle über Kriegsgerät erhält.

Abschließend kann festgehalten werden, dass sich Software prinzipiell in allen Bereichen einsetzen lässt. Von der simplen Automatisierung von Aufgaben wie beispielsweise dem Priorisieren von bestimmten Anwendungen (Zielerfassung, Raketensteuerung) für den Zugriff auf Sensoren oder Rechenkapazität bis hin zu neuen Aufgabenbereichen wie die Systemadministration im Einsatzgebiet ist das mögliche Spektrum sehr groß.

Allerdings ist es schwer, abzuschätzen, in welchen Bereichen das Militär wie weit auf Software zurückgreift, da die genauen Arbeitsabläufe strikter Geheimhaltung unterliegen und dieser Report nur öffentlich zugängliche Entwicklungen im Softwarebereich heranziehen konnte. Somit ist es möglich, einzuschätzen, wie präzise beispielsweise Zielerfassungsalgorithmen sein können. Doch ob

diese neueren Entwicklungen auch schon eingesetzt werden, ist unklar. Wir wissen weder, wie aktuell beispielsweise die Software auf Raketen gehalten wird, noch ob sich die Firmware solcher Raketen überhaupt aktualisieren lässt oder gleich ganz neue Raketen beschafft werden müssen.

Eine weitere Schwierigkeit beim Erstellen dieses Research Reports war zweifellos die bislang nicht vorhandene Systematisierung von Software mit Bezug auf mögliche militärische Anwendungsszenarien. Die bereits herausgearbeiteten Kategorien der softwaregestützten Informationsgewinnung sind sicherlich nicht die finalen, sondern müssen mit Blick auf die militärischen Anwendungsmöglichkeiten noch verfeinert und vertieft werden. Nichtsdestotrotz können sie bereits einen ersten Überblick über die verschiedenen Bereiche geben, in welchen Software militärisch genutzt wird.

Viele der Erkenntnisse hier sind nur ein Zwischenstand, auf deren Basis nachfolgende Analysen mehr in die Tiefe gehen können. Noch offene Fragen betreffen unter anderem Arbeitsdefinitionen – wie lässt sich militärisch nutzbare Software einteilen und abgrenzen? Bedenkt man, dass sich dieselben Kommunikationskanäle sowohl für *situational awareness* als auch für Command & Control eignen, stellt sich hier bereits ein Definitionsproblem. Daran anknüpfend ist eine weitere wichtige Frage, wie sich Rüstungskontrolle mit Bezug auf Software operationalisieren lässt und nach welchen Kriterien sich militärische Software in Zukunft bewerten lassen könnte. Hier spielen Einschätzungen nach der Proliferation eine Rolle, möglicherweise Wettrüsten um immer bessere Klassifizierer und ihre Modelle und inwiefern Software neue Arten von Krisen auslösen oder Regionen destabilisieren könnte.

Ebenfalls offen ist die Frage nach vor allem mittel- bis langfristigen Trends. Dadurch, dass dieser Research Report nur eine erste Analyse darstellt, konnte eine Einschätzung ferner in der Zukunft liegender Entwicklung bislang nicht geleistet werden. Außerdem konnten die konkreten Anknüpfungspunkte von Software an militärische Anwendungsszenarien bislang nicht ausgiebig untersucht werden.

Die militärische Verwendung von Softwaretechnologien sowie ihr Einfluss auf moderne Waffensysteme und die zukünftige Kriegsführung ist noch nicht hinreichend verstanden. Damit fehlen ein Bewusstsein um ihre friedens- und sicherheitspolitischen Risiken und Möglichkeiten ihrer Regulierung. Der vorliegende Research Report liefert einen Beitrag, um dieses Verständnis zu weiten und zukünftige Rüstungskontrollbemühungen zu unterstützen.

Endnoten

1 In diesem Artikel wird für militärische Kommando- und Kontrolleinrichtungen der englische Begriff „Command & Control“ verwendet, was durch den expliziten Fokus auf das Militär der USA begründet ist.

2 Es gibt beispielsweise den „HOW MANY HIPPOS“-Algorithmus belgischer Wissenschaftler, ein Infrarot-Zielerfassungsalgorithmus, der genutzt wird, um Nilpferde zu überwachen, vgl. Lhoest et al. 2015.

3 Die verwendeten Präfixe Kilo, Mega, Giga, Tera und Peta sind jeweils um einen 1.024er-Schritt größer als die vorigen, d.h. 1 Petabyte = 1.024 Terabyte; 1 Terabyte = 1.024 Gigabyte, 1 Gigabyte = 1.024 Megabyte, 1 Megabyte = 1.024 Kilobyte und 1 Kilobyte = 1.024 Byte.

4 Wortwörtlich „vom Ladenregal“. Ein Beispiel könnte die Verwendung der Tabellenkalkulationssoftware Excel für Berechnungen von Abschusspositionen für Artillerie sein.

5 Als „Open Source“ wird Software bezeichnet, deren Quellcode öffentlich einsehbar und dadurch analysierbar ist. Demgegenüber steht sogenannte „Closed Source“-Software, deren Quellcode nicht veröffentlicht wird und welche dadurch nicht ohne weiteres analysierbar ist.

6 „Sharding“ bezeichnet Methoden, um große Datenmengen auf eine solche Art und Weise in kleinere Stapel („batches“) aufzuteilen, dass die einzelnen Datenpunkte trotz ihrer Fragmentierung schnell gefunden werden können.

7 Ein Framework bezeichnet im Softwarebereich eine Sammlung bereits vorgefertigter Werkzeuge zur Entwicklung von Anwendungen. Beispielsweise bietet das für Webseiten entwickelte Framework „React“ bereits Funktionalität für Formulare (zum Beispiel für die Validierung von Eingabe-Feldern wie Passwörtern), sodass diese Logik nicht für jedes Projekt erneut implementiert werden muss.

8 Tatsächlich erklärt die U.S. Army in ihrem Field Manual Nr. 6-0: „Planners ensure that deployed INFOSYS implement open, nonproprietary, commonly accepted standards and protocols to interface with nonmilitary systems.“ (U.S. Army 2003: 5-14)

9 Im Kontext dieses Papers wird als „eingebettes System“ bzw. embedded system ein solches verstanden, welches autark, d.h. ohne Verbindungskabel zu Internet oder Stromnetz agieren kann.

10 Dies zeigt sich angesichts der COVID-19-Pandemie im Jahr 2020. Der DE-CIX, einer der weltweit größten Internet-Knotenpunkte, zeigt sich unbeeindruckt vom steigenden Datenverkehr, da auch die auf über 1 Terabyte pro Sekunde angestiegenen Datenraten durch die Infrastruktur gedeckt sind und noch rund 25 Prozent Reservekapazitäten vorhanden sind (vgl. Kannenberg 2020). In einer FAQ-Sektion bezüglich COVID-19 erklären die Betreiber: „Even if all companies in Europe were to operate exclusively remotely with staff all working from home, and the UEFA European Football Championship were to be broadcast in parallel, we would still be able to make the necessary bandwidth available for seamless interconnection.“ (vgl. DE-CIX 2020).

11 Umgangssprachlich besser bekannt als MPEG-4.

12 „Legacy“ (engl. für Erbe) bezeichnet in der Informationswissenschaft ein veraltetes, aber noch genutztes System.

13 Der Begriff „Geräusche“ bzw. „Noise“, der auch für visuelle Daten benutzt wird, stammt vom störenden Rauschen bei fehlendem Signal, welches aus älteren Radios zu hören ist, wenn nicht die richtige Frequenz eingestellt ist.

14 Als „terminal illumination“ wird die letzte Phase eines Lenkraketenfluges bezeichnet, bei welcher das anvisierte Ziel mit Radarstrahlen beschossen wird und somit aus der Perspektive des Sensors auf der Rakete „aufleuchtet“, was es der Rakete einfacher macht, das Ziel zu treffen.

15 Die Zahlen schwanken je nach Messmethode und auch pro einzelner Messung, da sie von zahlreichen Faktoren abhängig sind. Allerdings ist es eindeutig, dass die zentralen Recheneinheiten (CPU oder GPU) mit Abstand mehr Energie benötigen, als Arbeitsspeicher (RAM) oder Festplatten (insbesondere SSDs). Vgl. beispielsweise die Aufstellung auf der US-Hardware-Webseite *Tom's Hardware*: https://www.tomshardware.com/reviews/geforce-radeon-power,2122-7.html.

16 Ein Überblick über viele verschiedene TDLs und ihre Spezifikationen kann auf der SIGID-Wiki gefunden werden: https://www.sigidwiki.com/wiki/Category:Military

17 Vergleiche bspw. https://help.encoding.com/knowledge-base/article/understanding-bitrates-in-video-files/. Es sollte angemerkt werden, dass zahlreiche Faktoren die effektive Menge an Daten beeinflussen, darunter die Auflösung, die Kompressionsrate, die Bildwiederholrate (Framerate) und die Abtastrate des Bildsensors (Bittiefe). Gemäß der Protokollspezifikationen kann der H.264-Codec beispielsweise eine Datenmenge zwischen 64 Kilobits und 240 Megabits produzieren (vgl. Wiegand et al. 2003, S. 573, ebenso die offizielle MPEG-Seite: https://mpeg.chiariglione.org/faq/what-are-bitrates-are-supported-mpeg-4-visual).

18 GitHub ist eine Webseite, auf welcher kostenfrei Open Source-Software zum Download angeboten werden kann. Mittlerweile hat GitHub für quelloffene Software eine Monopolstellung erreicht.

19 Das Erkennen eines F-16-Jets oder das Erkennen einer Boeing 747 auf einem Bild sind einzelne Probleme, die zur Klasse „Erkennung von Flugzeugen auf Bildern" gehören.

20 Verschiedene Architekturen eignen sich für verschiedene Aufgabenbereiche, z.B. haben sich sogenannte „recurrent neural networks" (RNN) für Spracherkennung durchgesetzt und „convolutional neural networks" (CNN) für Bilderkennung.

21 Vgl. https://www.defense.gov/Newsroom/Contracts/Contract/Article/1999639/.

22 Implementations-agnostisch bedeutet hier, dass der Algorithmus ohne konkrete Implementation erklärt wird, d.h. anstelle tatsächlichem Computercodes erklärt er die notwendigen Schritte wie ein Kochrezept.

23 Azure ist die Produktbezeichnung der Cloud-Computing-Infrastruktur von Microsoft.

Glossar

Dieses Glossar fasst viele der im Report angesprochenen und verwendeten Begriffe in kompakter Form zusammen.

AEGIS: Benannt nach dem Schild Aigis aus der griechischen Mythologie ist das AEGIS Combat System ein integriertes Frühwarnsystem, das auf militärischen Schiffen verwendet wird, um sowohl feindliche Bedrohungen frühzeitig zu erkennen als auch Gegenmaßnahmen zu ergreifen. Es verfügt über → TDL-Schnittstellen, um von Kampfflugzeugen wie F-16-Jets zusätzliche Informationen zu erhalten, die somit effektiv das System erweitern.

Algorithmus: Abgeleitet vom persischen Gelehrten al-Chwarizmi, bezeichnet eine Reihe von Schritten, die ausgeführt werden müssen, um von einer Eingabe (Input) zu einer Ausgabe (Output) zu kommen. Beispiele sind händische Rechenoperationen wie Division, oder das Lösen eines „Rubik's Cube". Computer können ausschließlich mit Algorithmen arbeiten.

API: Application Programming Interface, eine Schnittstelle, welche es Programmen erlaubt, von anderen Programmen gesteuert zu werden. Ein Beispiel ist der Login-Prozess auf Internetdiensten, der entweder mittels einer grafischen Benutzeroberfläche (GUI) von menschlichen Benutzer*innen durchgeführt werden kann, oder aber von einem Programm voll automatisiert über eine solche API durchgeführt werden kann.

AWS (Amazon Web Services): Genau wie → Microsoft Azure ein Cloud Computing Service.

Big Data: Ein diffuser Begriff, welcher vornehmlich in Unternehmenskontexten verwendet wird, aber auch Einzug in wissenschaftliche Diskurse gefunden hat. Es gibt keine definitive Grenze, ab wann ein Datensatz „big" ist.

Binärzahlen: Die binäre Schreibweise ist eine andere Form, Zahlen zu repräsentieren als Dezimalzahlen, die nur mit zwei Ziffern (= binär) auskommt. Beispielsweise ließe sich die Zahl 4 als 100 und die Zahl 5 als 101 im Binärformat darstellen.

Code.mil: Die Haupt-Internetpräsenz der Open Source-Anstrengungen des US-Militärs.

Codec: Kleines Programm, welches auch direkt in Hardwarekomponenten integriert werden kann und auf Geschwindigkeit optimiert ist. Es wird genutzt für genutzt wird für extrem aufwändige Rechenoperationen wie das ver- und entschlüsseln von Videodaten.

Computerforensik: Eine Subdisziplin der Computerwissenschaften. Ähnlich forensischen Kriminologen suchen Computerforensiker nach Spuren von bestimmten Handlungen auf Computern oder in Software; beispielsweise nach Spuren gelöschter Emails oder bestimmten Dateien, die Rückschlüsse zulassen.

Convolutional Neural Network (CNN): Bekannteste Familie von modellbasierten Klassifizierern; ist in der Lage, mittels „Konvolution", d.h. mit Kontextinformationen Bilder zu klassifizieren.

CPU: Central Processing Unit, die flexibelste und älteste Form eines Rechenwerks in Computern.

Data Fusion: Beschreibt den Prozess der Fusion mehrerer Datenquellen zu einer. Dies kann entweder direkt nach dem Abschöpfen der Daten am Sensor selbst sein, d.h. am Beginn der Datenverarbeitungskette, wie es bei → LiDAR-Sensoren geschieht, oder am Ende der Datenverarbeitungskette, wenn nicht nur „rohe" Sensordaten zusammengeführt werden sollen, sondern auch bereits vorverarbeitete Daten, beispielsweise beim → Sharding oder → MapReduce. Im ersten Fall bezeichnet man die Data Fusion als Sensor Fusion, im zweiten als Information Integration.

Datenbank-ID: Eine eindeutige Nummer, welche einen Eintrag in einer Datenbank einzigartig beschreibt, ähnlich der Zeilennummern in Excel.

Dezimalzahl: Das geläufige Darstellungsformat für Zahlen mithilfe der zehn arabischen Ziffern von 0 bis 9.

Drift-Bias Error: Bezeichnet im Kontext von → Positionssensoren die Summierung von kleineren Messfehlern. Ein → INS-Sensor kann niemals die eigene Positionsänderung perfekt messen; es gibt immer einen kleinen Messfehler. Dadurch, dass ein INS-Sensor allerdings nur relative Daten misst, verstärkt sich dieser kleine Messfehler mit jeder neuen Messung. Beispielsweise, wenn die Position pro Messung um fünf Millimeter von der tatsächlichen abweicht, beträgt die Abweichung nach fünf Messungen bereits 2,5 Zentimeter.

Embedded System: Eingebettetes System; bezeichnet eine Vielzahl verschiedener Systeme ohne konkrete Grenze. Im Kontext dieses Papers wird als „eingebettes System" ein solches verstanden, welches autark, d.h. ohne Verbindungskabel zu Internet oder Stromnetz agieren kann und ausschließlich mittels Funkverbindungen kommuniziert.

Field of View (FOV): Technische Bezeichnung für den Bildausschnitt, den beispielsweise eine Kamera aufnehmen kann.

FLIR/SLIR: Forward Looking Infrared und Side-Looking Infrared; bezeichnet Infrarotsensoren, die entweder nach vorne an einer Waffenplattform angebracht sind, oder nach unten/zur Seite.

FPGA: Field Programmable Gate Arrays, ein besonderes Rechenwerk, welches sich von herkömmlichen → CPU, → GPU oder → TPU dadurch unterscheidet, dass es nicht eine Rechenoperation nach der anderen durchführen kann, sondern dass es kontinuierlich rechnen kann. Der Vorteil ist eine immense Geschwindigkeit, allerdings auf Kosten der Flexibilität, denn jedes FPGA muss für einen bestimmten Einsatzzweck programmiert werden und kann bis zu einer neuen Programmierung nicht für andere Zwecke genutzt werden.

Fullspectral Imagers: Alternative Bezeichnung für → Hyperspectral Imagers.

GPS: Global Positioning System, vgl. → Positionssensor.

GPU: Graphical Processing Unit, ein auf parallele Berechnungen von Gleitkommazahlen ausgelegtes Rechenwerk.

Graph: Ein Graph ist ein Begriff aus der Netzwerktheorie, der die Gesamtheit von Verbindungen zwischen verschiedenen Knotenpunkten bezeichnet, wobei beispielsweise die Knotenpunkte Personen und die Verbindungen Freunde, Familie und berufliche Kontakte sein können.

Heuristik: Eine Heuristik ist der Versuch, mit wenig Wissen zu wahrscheinlich guten Lösungen zu gelangen; vor allem im Bereich → NP-vollständiger Probleme werden diese benötigt, um Algorithmen schneller zu gestalten. Eine Heuristik sorgt dafür, dass nicht alle (potenziell unbegrenzt viele) Lösungswege abgesucht werden müssen, um eine optimale Lösung zu finden, sondern Abkürzungen genutzt werden können. Eine solche Heuristik ist beispielsweise für Zielerfassungsalgorithmen in Raketen die Annahme, dass helle Objekte im Bild Kampfflugzeuge sind sowie dass auch Täuschkörper auf dem Bild zu sehen sein könnten.

Hyperspectral Imagers: Sensoren, welche nicht nur das sichtbare oder Infrarot-Spektrum des Lichts messen können, sondern weit mehr Anteile, wodurch sie nicht nur in der Lage sind, Bilder zu erzeugen, sondern volumetrische „Bildwürfel“ mit einem Bild pro Licht-Spektrum.

Information Integration: vgl. → Data Fusion.

Infrarot-Sensor: Infrarot-Sensoren sind in der Lage, Infrarotstrahlen zu messen, die vor allem durch Wärmeabstrahlungen erzeugt werden.

INS: Inertial Navigation System, vgl. Positionssensor.

Integrated System: Ein integriertes System bezeichnet einen Zusammenschluss von Software, Hardware, in einigen Definitionen auch Menschen, welche zusammen effizienter oder präziser arbeiten können als die Einzelkomponenten für sich. Ein simples Beispiel ist die Integration der einzelnen Microsoft-Office-Produkte miteinander: Durch direktes Kommunizieren von Daten zwischen Excel und Word können Nutzer mit beiden Produkten effektiver arbeiten, als wenn die Daten händisch zwischen den beiden Programmen ausgetauscht werden müssten. Teilweise wird „System of Systems“ synonym verwendet; es gibt keine Anhaltspunkte, was konzeptuell am „System of Systems“-Ansatz des US-Militärs anders als an integrierten Systemen ist. Systemintegration bezeichnet die Handlung, welche disparate Systeme ineinander integriert.

Jamming: Bezeichnet den Vorgang, Funksignale zu stören, indem sogenanntes „weißes Rauschen“ erzeugt wird. Ein herkömmliches Radiosignal ließe sich beispielsweise jammen, indem auf derselben Frequenz (z.B. 106,7 MHz FM) ein gleichmäßiges Signal erzeugt wird. Das Radio gibt dann statt des Radioprogramms nur ein Rauschen aus. Im militärischen Kontext wird mit Jamming das Ziel verfolgt, die Kommunikation, aber auch – im Falle von Radar – die Erkennung von Objekten im Umkreis um die Radaranlage zu stören.

JEDI-Project: *Joint Enterprise Defence Infrastructure Project*, der Versuch des US-Militärs, an großflächige Cloud-Infrastruktur zu gelangen, mithilfe derer die erwarteten großen Datenmengen

der kommenden Jahre verarbeitet werden sollen. Nach einer initialen Zusage an → AWS wurde der Vertrag im Oktober 2019 an Microsoft → Azure vergeben, was durch AWS moniert wurde.

Kálmán-Filter: 1960 entwickelter Algorithmus, der in der Lage ist, Messfehler durch das Zusammenführen verschiedener, unabhängiger Datenquellen auszugleichen und somit die Signalqualität zu verbessern.

Kamera-Auflösung: Die Menge an Pixeln, welche eine Kamera erfassen kann. Bei normalen optischen Kameras sind dies heutzutage meist mindestens Full-HD, d.h. 1.920 x 1.080 Pixel. Infrarotkameras verfügen meist nur über eine Auflösung von 640 x 512 Pixeln.

LADAR: Siehe → LiDAR.

Landsat: Eine Familie von Satelliten, die für geographische Zwecke benutzt werden und die Erdoberfläche mittels verschiedener Sensoren kartographieren können.

Lethal Autonomous Weapon Systems (LAWS): Lethale autonome Waffensysteme; ein Begriff, der im Kontext der Convention on Certain Conventional Weapons (CCW) benutzt wird, um Drohnen wie die Predator oder Reaper unter ein Waffenkontrollregime zu setzen.

LiDAR: Abkürzung für Light Detection and Ranging; bezeichnet einen Sensor, welcher einen Laserstrahl aussendet und aus der Zeit, bis der reflektierte Strahl wieder am Sensor ankommt, die Entfernung zum Reflexionspunkt berechnet. Verwendet meist auch Daten von → Positionssensoren, um nicht nur die Entfernung zwischen Reflexionspunkt und Sensor zu berechnen, sondern auch die absolute Position des Reflexionspunktes.

Line-of-Sight (LoS): Militärischer Ausdruck für Kommunikation, die direkt stattfindet (d.h. nicht zwangsläufig mit Sichtkontakt).

Machine Learning: Bezeichnet den Vorgang, bei welchem nicht ein Algorithmus entwickelt wird, welcher eine bestimmte Aufgabe zuverlässig erfüllt, sondern ein Algorithmus mit Trainingsdaten ausgestattet wird, um ein sogenanntes → Modell zu erstellen, mit welchem der Algorithmus im Anschluss ähnliche Probleme zuverlässig lösen kann.

MEMS: Micro-Electromechanical Sensor, bezeichnet eine besonders kleine Ausführung eines Sensors; meist kleiner als eine Fingerkuppe. Vor allem Positionssensoren wie → GPS oder → INS werden heute vielfach als MEMS produziert und passen damit beispielsweise in Smartphones.

Microsoft Azure: Ein Cloud Computing Service, welcher entfernte Rechenoperationen ermöglicht, d.h. mittels → API werden Rechenoperationen an Serverfarmen gesendet, deren Ergebnisse im Anschluss an die Berechnung wieder an den Auftraggeber zurück gesendet werden.

Modell: Ein „Modell“ bezeichnet in der Computerwissenschaft eine Repräsentation von etwas. Dies kann entweder eine Silhouette eines erfassten Ziels bei Zielerkennungsalgorithmen sein, oder eine Matrix mit Zahlen (sogenannten Gewichten), die von modellbasierten Klassifizierern genutzt wird, um auf Bildern bestimmte Objekte zu erkennen.

Modellbasierter Klassifizierer: Ein Algorithmus, der in der Lage ist, mittels eines Modells Objekte in Bildern zu erkennen. Es gibt auch Klassifizierer für Sprache und Text. Teilweise werden solche Klassifizierer sogar zur Reinigung von Daten verwendet.

Neuronales Netzwerk: Gängige Bezeichnung für → modellbasierte Klassifizierer.

O-Notation: Mit der O-Notation wird die Laufzeit von Algorithmen im schlimmsten Fall (Worst Case) angegeben, wobei ein n^2 generell bedeutet, dass die Laufzeit mit steigenden Rechenoperationen (d.h. n) exponentiell ansteigt und Optimierungen versuchen, die Laufzeit auf log(n) zu begrenzen, da die Laufzeit mit steigendem n nicht mehr exponentiell wächst.

Off-the-Shelf-Software: Wörtlich übersetzt „vom Ladenregal“, militärische Bezeichnung für Software, die nicht dezidiert für militärische Zwecke entwickelt wurde, aber dennoch in militärischen Kontexten einsetzbar ist.

Open Source: Bezeichnet Software, deren Quellcode öffentlich zugänglich ist; meist unter einer liberalen Lizenz, sodass andere die Software kostenlos nutzen und weiter entwickeln können. Im Kontext des Militärs dient Open Source dazu, kostengünstig an verbesserte Algorithmen und Softwarepakete zu gelangen.

Over-the-Horizon (OTH): Militärischer Ausdruck für Kommunikation, die über die Erdkrümmung hinweg stattfindet. Wird einerseits bei Raketenwarnsystemen verwendet, deren Radarwellen mit der Erdkrümmung wandern, oder indirekte Kommunikation, beispielsweise über Satelliten.

Perzeptron: Bezeichnung für die ersten Klassifizierer, die heute als „neuronale Netzwerke" bekannt sind.

Positionssensor: Ein Positionssensor ist in der Lage, die eigene Position in Relation zur Erde absolut oder relativ zu bestimmen. Die zwei wichtigsten Sensoren sind → GPS und → INS. Das GPS ist in der Lage, mittels einem Netzwerk von Satelliten die eigene Position absolut zu bestimmen, da die Satelliten geostationär sind, d.h. sie verändern ihre Position nicht, welche bekannt ist. Das INS wiederum ist in der Lage, die Position relativ zu bestimmen (d.h. inertial). Das INS besteht eigentlich aus mehreren Sensoren, verwendet also im Grunde eine Form von → Sensor Fusion, da es einen Magnet-Sensor (Kompass), einen Rotations-Sensor (Gyroskop) und einen Luftdruck-Sensor (Barometer) verwendet, um Positionsänderungen zu errechnen. GPS-Sensoren können aufgrund ihrer Abhängigkeit von Satelliten sowohl → gespooft als auch → gejammt werden, während die Achillesferse der INS-Sensoren der → Drift-Bias Error ist.

RADAR: Abkürzung für Radio Detection and Ranging; bezeichnet einen Sensor, welcher mithilfe von Radiowellen Objekte aufspüren und deren Distanz, Form und Geschwindigkeit erfassen kann.

Sensor Fusion: vgl. → Data Fusion.

Sensor: Ein Sensor wandelt Informationen der physikalischen Welt in digitale Daten um. Beispiele sind Lichtreflexionen, Geräusche, Luftdruck, die eigene Position oder Geographie. Sensoren werden benötigt, um Computern dabei zu helfen, die Umgebung zu erfassen.

Signal-to-Noise-Ratio (SNR): Anteil von unerwünschtem „Grundrauschen" in einer Signalquelle; je geringer dieser Anteil, desto schärfer bzw. klarer ist ein Signal. Beim → Jamming wird versucht, die SNR zu maximieren, sodass auch gute Algorithmen nicht mehr in der Lage sind, das Signal aus dem Rauschen herauszufiltern.

SIMDIS SDK: Ein „Software Development Kit" (SDK) der US-Navy, welches in der Lage ist, arbiträre Daten zu verarbeiten und in einem einzigen, benutzerdefinierten System darzustellen. SIMDIS SDK funktioniert nicht einfach mittels Download, sondern stellt vielmehr zahlreiche Funktionen zur Verfügung, welche von Programmierern genutzt werden können, um ein benutzerdefiniertes System zu entwickeln, welches für bestimmte Einsatzszenarien verwendet werden kann.

Spoofing: Unter „spoofing" versteht man in der Informationswissenschaft das Fälschen von Informationen. Spoofing wird im militärischen Kontext verwendet, um den Feind zu täuschen, indem beispielsweise die eigene Streitkraft auf den Radargeräten des Feindes größer oder kleiner erscheint als sie eigentlich ist. Im Kontext von GPS-Sensoren bezeichnet Spoofing den Versuch, den Sensor mit falschen Daten zu täuschen, sodass er von einer anderen Position ausgeht, als die tatsächliche.

SSD: Solid State Disk, eine moderne Form des Datenspeichers, welches statt älterer Festplatten (Hard Disk Drive, HDD) auf Flash-Speicher setzt, welche weitaus schneller und stromsparender funktionieren, da sie keine beweglichen Teile besitzen.

StarLink: Ein Satellitenprogramm des US-Konzerns SpaceX, dessen Ziel es ist, mittels tausender kleiner Satelliten ein erdumspannendes Internet-Netzwerk aufzuspannen.

SWaP-Restriktion: Als „Size, Weight, and Power"-Restriktion wird insbesondere im Kontext von ferngesteuerten Systemen wie Drohnen und Robotern bezeichnet, dass diese über beschränk-

te Ressourcen (im Vergleich zu fest installierten Computersystemen) verfügen. Beispielsweise verfügen Drohnen über begrenzte Treibstoff- und Batteriereserven und können nicht unbegrenzt schwere oder besonders große Komponenten aufnehmen.

Synthetic-Aperture / Fixed-Aperture: Bezeichnet zwei Bauformen von Radargeräten. Während bei fixed-aperture-Radars die Radaranlage physisch bewegt, d.h. rotiert werden muss, können synthetic-aperture-Radars ihr Signal elektrisch bewegen, was die Verschleißerscheinungen reduziert und auch den Platz, den solche Radargeräte verwenden.

System of Systems: vgl. → Integrated System.

Systemintegration: vgl. → Integrated System.

Tactical Data Link (TDL): Militärische Bezeichnung für Funkkommunikation, die besonderen militärischen Sicherheitsansprüchen genügen.

TPU: Tensor Processing Unit, ein auf die Berechnung ganzer Matrizen (Tensoren) ausgelegtes Rechenwerk des US-Unternehmens Google.

Uplink: Bezeichnung für eine Datenverbindung. Wird auch als Gegenbegriff zu Downlink verwendet, um Daten zu beschreiben, die von einem Computer an ein entferntes Ziel gesendet werden. byte = 1.024 Megabyte, 1 Megabyte = 1.024 Kilobyte und 1 Kilobyte = 1.024 Byte.

Literatur

Adão, Telmo, Hruška , Jonáš, Pádua, Luís, Bessa, José, Peres, Emanuel, Morais, Raul & Sousa, Joaquim (2017) Hyperspectral Imaging: A Review on UAV-Based Sensors, Data Processing and Applications for Agriculture and Forestry. Remote Sensing 9(11): 1–30. https://doi.org/10.3390/rs9111110.

ADLink (2019): Tower in a Teacup: How the Small Form Factor Transition Is Reshaping Embedded and Military Computing. Online verfügbar unter http://go.adlinktech.com/NCP02-2019_WP_Small-Form-Factor_DL.html (abgerufen am 28. April 2020).

Aizman, Alex, Maltby, Gavin & Breuel, Thomas (2019). High Performance I/O for Large Scale Deep Learning. IEEE International Conference on Big Data 2019: 1–6.

Alberts, David (1999). Network Centric Warfare: Developing and Leveraging Information Superiority. o.O.: DoD C4ISR Cooperative Research Program.

Alonzo, Mark (2013). Did You Say Lidar or Ladar? In Harris Geospatial Blog. Online verfügbar unter https://www.harrisgeospatial.com/Learn/Blogs/Blog-Details/ArtMID/10198/ArticleID/16391/Did-you-say-LiDAR-or-LADAR (abgerufen am 28. April 2020).

Alwardt, Christian & Polle, Johanna (2018). Internationale Rüstungskontrollbemühungen zu Autonomen Waffensystemen: Definitionen, Technik und sicherheitspolitische Implikationen. Sicherheit und Frieden 36(3): 133–139. https://doi.org/10.5771/0175-274X-2018-3-133.

Anderson, E. S., Thompson, James A. & Austin, R. E. (2005). LIDAR Density and Linear Interpolator Effects on Elevation Estimates. International Journal of Remote Sensing 26(18): 3889–3900. https://doi.org/10.1080/01431160500181671.

Bas, Nuray, Coskun, H. Gonca, Kaya, Sinasi, Bayram, Bulent & Celik, Hakan (2018). Effective Lidar Data Classification by Row Data and Parameter Analysis Framework. Fresenius Environmental Bulletin 27(6): 4068–4075. https://www.researchgate.net/profile/Nuray_Bas/publication/327800752_Effective_LiDAR_data_classification_by_row_data_and_parameter_analysis_framework/links/5d088ace299bf1f539cb9fd3/Effective-LiDAR-data-classification-by-row-data-and-parameter-analysis-framework.pdf (abgerufen am 11. Juni 2020).

Bauer, Meredith Rutland (2018). The Security Solution Found Inside Shipping Containers. In The Fifth Domain. Online verfügbar unter https://www.fifthdomain.com/dod/2018/02/23/the-security-solution-found-inside-shipping-containers/ (abgerufen am 28. April 2020).

Boulanin, Vincent & Verbruggen, Maaike (2017). Mapping the Development of Autonomy in Weapon Systems. Stockholm: SIPRI.

Callahan, Devon, Shakarian, Paulo, Nielsen, Jeffrey & Johnson, Anthony N. (2012). Shaping Operations to Attack Robust Terror Networks. IEEE International Conference on Social Informatics: 13–18. https://doi.org/10.1109/SocialInformatics.2012.22.

Cao, Yutian, Wang, Gang, Yan, Dongmei und Zhao, Zhongming (2015). Two Algorithms for the Detection and Tracking of Moving Vehicle Targets in Aerial Infrared Image Sequences. Remote Sensing 8(1): 1–21. https://doi.org/10.3390/rs8010028.

Cechowicz, Radosław (2017). Bias Drift Estimation for Mems Gyroscope Used in Inertial Navigation. Acta Mechanica et Automatica 11(2): 104–110. https://doi.org/10.1515/ama-2017-0016.

Chamayou, Grégoire (2015). Drone Theory. London: Penguin.

Cole, Siobhan Gorman (2009). Insurgents Hack U.S. Drones. In Wall Street Journal. Online verfügbar unter https://www.wsj.com/articles/SB126102247889095011 (abgerufen am 28. April 2020).

Conger, Kate, Sanger, David & Shane, Scott (2019). Microsoft Wins Pentagon's $10 Billion JEDI Contract, Thwarting Amazon. In The New York Times. Online verfügbar unter https://www.nytimes.com/2019/10/25/technology/dod-jedi-contract.html (abgerufen am 29. April 2020).

Crick, Francis (1989). The Recent Excitement About Neural Networks. Nature 337(6203): 129–132. https://doi.org/10.1038/337129a0.

Crouse, David Frederic (2017). The Tracker Component Library: Free Routines for Rapid Prototyping. IEEE Aerospace and Electronic Systems Magazine 32(5): 18–27. https://doi.org/10.1109/MAES.2017.160215.

Crowe, Steve (2019). Researchers back Tesla's non-LiDAR approach to self-driving cars. In The Robot Report. Online verfügbar unter https://www.therobotreport.com/researchers-back-teslas-non-lidar-approach-to-self-driving-cars/ (abgerufen am 28. April 2020).

Davis, Zachary S. (2019). Artificial Intelligence on the Battlefield. An Initial Survey of Potential Implications for Deterrence, Stability, and Strategic Surprise. Center for Global Security Research. Online verfügbar unter https://cgsr.llnl.gov/content/assets/docs/CGSR-AI_BattlefieldWEB.pdf (abgerufen am 19. Mai 2020).

DE-SIX (2020). FAQs on the COVID-19 situation. Online verfügbar unter https://www.de-cix.net/en/about-de-cix/company-profile/faqs-on-covid-19-situation (abgerufen am 08. April 2020).

Dean, Jeffrey & Ghemawat, Sanjay (2008). MapReduce: Simplified Data Processing on Large Clusters. Communications of the ACM 51(1): 107–113. https://doi.org/10.1145/1327452.1327492.

Defense Acquisition University (2001). System Engineering Fundamentals. Fort Belvoir: Defense Acquisition University Press.

Deng, Jia, Dong, Wei, Socher, Richard, Li, Li-Jia, Li, Kai & Fei-Fei, Li (2009). ImageNet: A Large-Scale Hierarchical Image Database. IEEE Conference on Computer Vision and Pattern Recognition. https://doi.org/10.1109/CVPR.2009.5206848.

Elmenreich, Wilfried (2002). An Introduction to Sensor Fusion. Research Report 47/2001.: Wien: Institut für Technische Informatik. https://www.researchgate.net/profile/Wilfried_Elmenreich/publication/267771481_An_Introduction_to_Sensor_Fusion/links/55d2e45908ae0a3417222dd9.pdf (abgerufen am 11. Juni 2020).

Elsayed, Gamaleldin F., Goodfellow, Ian & Sohl-Dickstein, Jascha (2019). Adversarial Reprogramming of Neural Networks. arXiv:1806.11146 [cs.LG]. https://arxiv.org/abs/1806.11146.

Erwin, Sandra (2017). Satellite Operators Push Plan to Upgrade Military Spy Drones. In Spacenews. Online verfügbar unter https://spacenews.com/satellite-operators-push-plan-to-upgrade-military-spy-drones/ (abgerufen am 28. April 2020).

Fairfax, Luisa D. & Fresconi, Frank E. (2012). Position Estimation for Projectiles Using Low-Cost Sensors and Flight Dynamics. Army Research Laboratory Report. O.O.: Army Research Laboratory.

Fasano, Giancarmine, Accardo, Domenico, Tirri, Anna Elena, Moccia, Antonio & de Lellis, Ettore (2015). Radar/electro-optical data fusion for non-cooperative UAS sense and avoid. Aerospace Science and Technology 46: 436–450.

Field, Kyle (2020). Tesla Replicated The Visibility Of Lidar With Its Realtime Vision Processing System. In Clean Technica. Online verfügbar unter https://cleantechnica.com/2020/04/24/tesla-achieved-the-accuracy-of-lidar-with-its-advanced-computer-vision-tech/ (abgerufen am 28. April 2020).

Haimovich, Alexander, Blum, Rick & Cimini, Leonard (2008). MIMO Radar with Widely Separated Antennas. IEEE Signal Processing Magazine 25(1): 116–129. https://doi.org/10.1109/MSP.2008.4408448.

Hassanien, Aboulnasr & Vorobyov, Sergiy A. (2010). Phased-Mimo Radar: A Tradeoff Between Phased-Array and Mimo Radars. IEEE Transactions on Signal Processing 58(6): 3137–3151. https://doi.org/10.1109/TSP.2010.2043976.

Hill, Kashmir (2020). The Secretive Company That Might End Privacy as We Know It. In The New York Times. Online verfügbar unter https://www.nytimes.com/2020/01/18/technology/clearview-privacy-facial-recognition.html (abgerufen am 28. April 2020).

The IEEE Global Initiative on Ethics of Autonomous and Intelligent Systems (2019). Ethically Aligned Design: A Vision for Prioritizing Human Well-being with Autonomous and Intelligent Systems. IEEE. Online verfügbar unter https://standards.ieee.org/content/ieee-standards/en/industry-connections/ec/autonomous-systems.html (abgerufen am 29. April 2020).

Instagram Engineering (2016). Sharding & Ids at Instagram. Online verfügbar unter https://instagram-engineering.com/sharding-ids-at-instagram-1cf5a71e5a5c (abgerufen am 28. April 2020).

Ishibashi, Shojiro, Tsukioka, Satoshi, Yoshida, Hiroshi, Hyakudome, Tadahiro, Sawa, Takao, Tahara, Junichro, Aoki, Taro & Ishikawa, Akihisa (2007). Accuracy Improvement of an Inertial Navigation System Brought About by the Rotational Motion. OCEANS Europe Konferenz: 1–5. https://doi.org/10.1109/OCEANSE.2007.4302282.

Javadnejad, Farid, Gillins, Daniel T., Parrish, Christopher E. & Slocum, Richard K. (2020). A Photogrammetric Approach to Fusing Natural Colour and Thermal Infrared Uas Imagery in 3D Point Cloud Generation. International Journal of Remote Sensing 41(1): 211–237. https://doi.org/10.1080/01431161.2019.1641241.

Jensen, Benjamin & Paschkewitz, John (2019). Mosaic Warfare: Small and Scalable are Beautiful. In War on the rocks. Online verfügbar unter https://warontherocks.com/2019/12/mosaic-warfare-small-and-scalable-are-beautiful/ (abgerufen am 08. April 2020).

Jiang, Liu, Li, Bing & Song, Meina (2010). THE Optimization of HDFS Based on Small Files. 3rd IEEE International Conference on Broadband Network and Multimedia Technology: 912–915. https://doi.org/10.1109/ICBNMT.2010.5705223.

Kang, Xudong, Zhang, Xiangping, Li, Shutao, Li, Kenli, Li, Jun & Benediktsson, Jon Atli (2017). Hyperspectral Anomaly Detection with Attribute and Edge-Preserving Filters. IEEE Transactions on Geoscience and Remote Sensing 55(10): 5600–5611. https://doi.org/10.1109/TGRS.2017.2710145.

Kannenberg, Axel (2020). DE-CIX: Steigender Datenverkehr Wegen Corona-Krise. In Heise. Online verfügbar unter https://www.heise.de/newsticker/meldung/DE-CIX-Steigender-Datenverkehr-wegen-Corona-4690509.html (abgerufen am 28. April 2020).

Kálmán, Rudolf E. (1960). A New Approach to Linear Filtering and Prediction Problems. Journal of Fluids Engineering 82(1): 35–45. https://doi.org/10.1115/1.3662552.

Kimmel, Troy S. (2009). Overview of AEGIS. Naval Engineers Journal 121(3): 27–35. https://doi.org/10.1111/j.1559-3584.2009.00202.x.

Knight, Will (2019). Military Artificial Intelligence Can Be Easily and Dangerously Fooled. In Technology review. Online verfügbar unter https://www.technologyreview.com/s/614497/military-artificial-intelligence-can-be-easily-and-dangerously-fooled/amp/ (abgerufen am 28. April 2020).

Lendon, Brad (2020). In 1988, a US Navy warship shot down an Iranian passenger plane in the heat of battle. In CNN Edition. Online verfügbar unter https://edition.cnn.com/2020/01/10/middleeast/iran-air-flight-655-us-military-intl-hnk/index.html (abgerufen am 29. April 2020).

Leon, Harmon (2019). Top Secret Military-Grade Surveillance Drones Might Be Coming to Your Neighborhood. In Observer. Online verfügbar unter https://observer.com/2019/06/gorgon-stare-aerial-surveillance-drones/ (abgerufen am 28. April 2020).

Lhoest, Simon, Linchant, Julie, Quevauvillers, Samuel, Vermeulen, Cédric & Lejeune, Philippe (2015). HOW Many Hippos (Homhip): Algorithm for Automatic Counts of Animals with Infra-Red Thermal Imagery from UAV. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XL-3/W3. https://doi.org/10.5194/isprsarchives-XL-3-W3-355-2015.

Longdon, Jason (2013). Link 22 Integration Needs Overview. Präsentation auf MILCIS 2013, Canberra. Online verfügbar unter https://static1.squarespace.com/static/5274112ae4b02d3f058d4348/t/529c59bfe4b07443d2dcb951/1385978303920/1-8c.pdf (abgerufen am 29. April 2020).

Loquercio, Antonio, Maqueda, Ana I., Blanco, Carlos R. & Scaramuzza, Davide (2018). DroNet: Learning to Fly by Driving. IEEE Robotics and Automation Letters 3(2): 1088–1095. https://doi.org/10.1109/LRA.2018.2795643.

Luger, George F. (2008). Artificial Intelligence. Structures and Strategies for Complex Problem Solving. Boston: Pearson Education.

Maathuis, B. H. P. & Genderen, J. L. (2004). A Review of Satellite and Airborne Sensors for Remote Sensing Based Detection of Minefields and Landmines. International Journal of Remote Sensing 25(23): 5201–5245. https://doi.org/10.1080/01431160412331270803.

Manolakis, Dimitris & Shaw, Gary (2002). Detection Algorithms for Hyperspectral Imaging Applications. IEEE Signal Processing Magazine 19(1): 29–43. https://doi.org/10.1109/79.974724.

Manthorpe, William H J. (1996). The Emerging Joint System of Systems: A Systems Engineering Challenge and Opportunity for Apl. Johns Hopkins APL Technical Digest 17(3): 305–313.

Martinez-Ruiz, Manuel, Artes-Rodriguez, Antonio, Diaz-Rico, Jose Antonio & Blanco Fuentes, Jose (2010). New Initiatives for Imagery Transmission over a Tactical Data Link. A Case Study: JPEG2000 Compressed Images Transmitted in a Link-16 Network. Method and Results. MILCOM 2010 Military Communications Conference: 1163–1168. https://doi.org/10.1109/MILCOM.2010.5680102.

Mazzucato, Mariana (2014). Das Kapital des Staates. Eine andere Geschichte von Innovation und Wachstum. München: Verlag Antje Kunstmann.

McDonough, Marty (2010). Tactical Data Links: Decades Old, but Still Talking with the Big Boys. In Military Embedded. Online verfügbar unter http://mil-embedded.com/articles/tactical-president-ceo-tactical-communications-group/ (abgerufen am 28. April 2020).

McLain, Christopher & King, Janet (2017). Future Ku-Band Mobility Satellites. 35th AIAA International Communications Satellite Systems Conference. https://doi.org/10.2514/6.2017-5412.

Meyer, Roland (2019). Operative Porträts: Eine Bildgeschichte Der Identifizierbarkeit von Lavater Bis Facebook. Konstanz: Konstanz University Press.

Münkler, Herfried (2004). Die Neuen Kriege. Reinbek: Rowohlt.

Oh, Seong Joon, Augustin, Max, Schiele, Bernt & Fritz, Mario (2018). Towards Reverse-Engineering Black-Box Neural Networks. arXiv:1711.01768 [stat.ML]. http://arxiv.org/abs/1711.01768.

Porche, Isaac (2014). data_flood. Helping the Navy address the Rising Tide of Sensor Information. Santa Monica: RAND.

Qu, Wenwen, Wang, Cheng, Wang, Hanyun, Cheng, Ming & Li, Jonathan (2014). Triple-Surface Structure Extraction and Fitting for Control Point Positioning in Lidar Point Clouds. Proceedings of the International Conference on Internet Multimedia Computing and Service: 136–139. https://doi.org/10.1145/2632856.2632934.

Rawnsley, Adam (2011). Iran's Alleged Drone Hack: Tough, but Possible. In Wired. Online verfügbar unter https://www.wired.com/2011/12/iran-drone-hack-gps/ (abgerufen am 28. April 2020).

Redmon, Joseph & Farhadi, Ali (2016). YOLO9000: Better, Faster, Stronger. arXiv:1612.08242 [cs.CV]. http://arxiv.org/abs/1612.08242.

Sato, Kaz, Young, Cliff & Patterson, David (2017). An in-Depth Look at Google's First Tensor Processing Unit (TPU). Online verfügbar unter https://cloud.google.com/blog/products/gcp/an-in-depth-look-at-googles-first-tensor-processing-unit-tpu/ (abgerufen am 28. April 2020).

Scharre, Paul & Horowitz, Michael C. (2015). An Introduction to Autonomy in Weapon Systems. Washington: Center for a New American Security.

Schmitt, Marwin, Redi, Judith, Cesar, Pablo & Bulterman, Dick (2016). 1Mbps Is Enough: Video Quality and Individual Idiosyncrasies in Multiparty Hd Video-Conferencing. Eighth International Conference on Quality of Multimedia Experience. https://doi.org/10.1109/QoMEX.2016.7498961.

Schröder, Thorsten & Buermeyer, Ulf (2019). 36C3 - Finfisher Verklagen. 36. Chaos Communication Congress. Online verfügbar unter https://www.youtube.com/watch?v=kvGIY0JYg0k (abgerufen am 28. April 2020).

Shi, Xinchu, Ling, Haibin, Blasch, E & Hu, Weiming (2012). Context-Driven Moving Vehicle Detection in Wide Area Motion Imagery. 21st International Conference on Pattern Recognition: 1–4.

SQLite (o.J.). 35% Faster Than The Filesystem. Online verfügbar unter https://www.sqlite.org/fasterthanfs.html. (abgerufen am 08. April 2020).

Swart, G. (2004). Spreading the Load Using Consistent Hashing: A Preliminary Report. Third International Symposium on Parallel and Distributed Computing: 169–176. https://doi.org/10.1109/ISPDC.2004.47.

Taghipour, Ashkan & Ghassemian, Hassan (2017). Hyperspectral Anomaly Detection Using Attribute Profiles. IEEE Geoscience and Remote Sensing Letters 14(7): 1136–1140. https://doi.org/10.1109/LGRS.2017.2700329.

Tippenhauer, Nils Ole, Pöpper, Christina, Bonne Rasmussen, Kasper & Capkun, Srdjan (2011). On the Requirements for Successful Gps Spoofing Attacks. Proceedings of the 18th ACM conference on Computer and communications security: 75–86. https://doi.org/10.1145/2046707.2046719.

U.S. Army (2003). Field Manual No. 6–3. Washington: Department of the Army. Online abrufbar unter https://www.bits.de/NRANEU/others/amd-us-archive/fm6(03).pdf. (abgerufen am 08. April 2020).

U.S. Naval Research Laboratory (2019). SIMDIS SDK Presentation. Online verfügbar unter https://simdis.nrl.navy.mil/SimdisPresentation.aspx (abgerufen am 28. April 2020).

Venieris, Stylianos I., Kouris, Alexandros & Bouganis, Christos-Savvas (2018). Deploying Deep Neural Networks in the Embedded Space. arXiv:1806.08616 [cs.CV]. http://arxiv.org/abs/1806.08616.

Venkateswaran, Narayanan & Changder, Suvamoy (2017). Simplified Data Partitioning in a Consistent Hashing Based Sharding Implementation. Proceedings of the 2017 IEEE Region 10 Conference, Malaysia: 895–900 https://doi.org/10.1109/TENCON.2017.8227985.

Wallace, Luke, Lucieer, Arko & Watson, Christopher S. (2014). Evaluating Tree Detection and Segmentation Routines on Very High Resolution UAV Lidar Data. IEEE Transactions on Geoscience and Remote Sensing 52(12): 7619–7628. https://doi.org/10.1109/TGRS.2014.2315649.

Wang, Yu Emma, Wie, Gu-Yeon & Brooks, David (2019). Benchmarking TPU, GPU, and CPU Platforms for Deep Learning. arXiv:1907.10701 [cs.LG] http://arxiv.org/abs/1907.10701.

Wang, Yan, Chao, Wei-Lun, Garg, Divyansh, Hariharan, Bharath, Campbell, Mark & Weinberger, Kilian Q. (2020). Pseudo-LiDAR from Visual Depth Estimation: Bridging the Gap in 3D Object Detection for Autonomous Driving. arXiv:1812.07179 [cs.CV]. https://arxiv.org/abs/1812.07179.

Wiegand, Thomas, Sullivan, Gary J., Bjontegaard, Gisle & Luthra, Ajay (2003). Overview of the H.264/Avc Video Coding Standard. IEEE Transactions on Circuits and Systems for Video Technology 13(7): 560–576. https://doi.org/10.1109/TCSVT.2003.815165.

Wu, Sijie, Zhang, Kai, Niu, Saisai & Yan, Jie (2019). Anti-Interference Aircraft-Tracking Method in Infrared Imagery. Sensors 19(6): 1–25. https://doi.org/10.3390/s19061289.

Yilmaz, Alper, Shafique, Khurram & Shah, Mubarak (2003). Target Tracking in Airborne Forward Looking Infrared Imagery. Image and Visual Computing 21(7): 623–635. https://doi.org/10.1016/S0262-8856(03)00059-3.

Yin, Dameng und Wang, Le (2019). Individual Mangrove Tree Measurement Using UAV-Based Lidar Data: Possibilities and Challenges. Remote Sensing of Environment 223(March): 34–49. https://doi.org/10.1016/j.rse.2018.12.034.

Yussupov, Vladimir, Soldani, Jacopo, Breitenbücher, Uwe, Brogi, Antonio & Leymann, Frank (2020). FaaSten Your Decisions: Classification Framework and Technology Review of Function-as-a-Service Platforms. arXiv:2004.00969 [cs.SE]. https://arxiv.org/abs/2004.00969.

Zhao, Rui, Du, Bo & Zhang, Liangpei (2017). Hyperspectral Anomaly Detection via a Sparsity Score Estimation Framework. IEEE Transactions on Geoscience and Remote Sensing 55(6): 3208–3222. https://doi.org/10.1109/TGRS.2017.2664658.

ÜBER DEN AUTOR

Hendrik Erz ist wissenschaftlicher Mitarbeiter im DSF-geförderten Projekt „Algorithmen und Künstliche Intelligenz als Game Changer?“ am Institut für Friedensforschung und Sicherheitspolitik an der Universität Hamburg (IFSH).

erz@ifsh.de

ÜBER DAS PROJEKT

Gefördert durch die Deutsche Stiftung Friedensforschung (DSF) untersucht das Projekt „Algorithmen und Künstliche Intelligenz als Game Changer? Moderne Waffensysteme zwischen Erwartung und Wirklichkeit“ die sicherheitspolitischen Implikationen von Softwaretechnologien.

gefördert durch
DSF Deutsche Stiftung Friedensforschung
german foundation for peace research

ÜBER DAS INSTITUT

Das Institut für Friedensforschung und Sicherheitspolitik (IFSH) erforscht die Bedingungen von Frieden und Sicherheit in Deutschland, Europa und darüber hinaus. Das IFSH forscht eigenständig und unabhängig. Es wird von der Freien und Hansestadt Hamburg finanziert.

Gefördert von:
Hamburg | Behörde für Wissenschaft, Forschung und Gleichstellung

IFSH – **Institut für Friedensforschung und Sicherheitspolitik** an der Universität Hamburg
Beim Schlump 83 20144 Hamburg Germany Phone +49 40 866077-0 ifsh@ifsh.de www.ifsh.de